Linguistic properties of multi-word passphrases

Joseph Bonneau, Ekaterina Shutova

Computer Laboratory
University of Cambridge
{jcb82,es407}@cl.cam.ac.uk

Abstract. We examine patterns of human choice in a passphrase-based authentication system deployed by Amazon, a large online merchant. We tested the availability of a large corpus of over 100,000 possible phrases at Amazon's registration page, which prohibits using any phrase already registered by another user. A number of large, readily-available lists such as movie and book titles prove effective in guessing attacks, suggesting that passphrases are vulnerable to dictionary attacks like all schemes involving human choice. Extending our analysis with natural language phrases extracted from linguistic corpora, we find that phrase selection is far from random, with users strongly preferring simple noun bigrams which are common in natural language. The distribution of chosen passphrases is less skewed than the distribution of bigrams in English text, indicating that some users have attempted to choose phrases randomly. Still, the distribution of bigrams in natural language is not nearly random enough to resist offline guessing, nor are longer three- or four-word phrases for which we see rapidly diminishing returns.

1 Introduction

Despite decades of research on the vulnerability of human-chosen passwords to guessing attacks [17], passwords continue to dominate web authentication. Passwords' familiarity and extremely low implementation costs are believed to be key reasons for their persistence [9], particularly given failures in the market for web authentication which discourage radical changes [4].

Given these constraints, multi-word passphrases may be a promising improvement, as they require few implementation changes and offer a similar user experience. Requiring multiple words in a password is a natural extension of usability findings which have suggested that mnemonic-phrase passwords[1] are considerably more difficult to guess while still easily memorable [22]. Recent research has also suggested that increasing the minimum length of passwords is the most effective means of increasing security in place of requirements to include character classes like numbers or symbols [12].

[1] Mnemonic-phrase passwords are formed by condensing a natural language sentence like "George Michael and Ann went to the protest on Friday" into a relatively-strong password like `GM&Aw2tpoF`.

Specific usability studies of passphrases [11] have found them to be just as memorable as passwords, subject to an increased rate of typographical errors. Several proposals have been made reduce the rate of errors, either by storing multiple hashes of a passphrase to recognise entry of nearly-correct strings [16,2] or by providing visual feedback to allow a user to notice typos when they are made [18]. Passphrases may in fact be more usable in the context of mobile phones, which have input interfaces optimised for natural language and not for pseudorandom character strings [10]. Passphrases are already deployed in widely-used PGP software to protect private keys on disk [23] which has led to speculative research on hardware brute-forcing attacks [21].

Still, the security gains of moving from simple passwords to passphrases are unknown. The few published usability studies of passphrases estimate security either by naive calculations of the total space of possible character strings [11] or rely on Shannon's decades-old estimates of the entropy of characters in English text [20]. Experience from password guessing suggests that the only valid methods of estimating security of human-chosen secrets like passphrases are to run cracking software against real choices [17] or to collect sufficient data that the frequency of common choices can be predicted statistically [3]. Kuo et al. assembled a dictionary of phrases to evaluate the strength of mnemonic-phrase passwords [14], but we are unaware of any attempt to conduct a guessing attack on real human-chosen passphrases.

In this work we study passphrase choices using data collected from the Amazon PayPhrase system. Launched in 2009 for customers in the USA only, this system allows users to register a passphrase to make web purchases and is one of the few passphrase schemes widely deployed on the Internet. While we don't have access to the entire corpus of registered phrases, we can identify general linguistic patterns in passphrase selection which have important implications for future research on passphrases.

2 Data collection

In the Amazon PayPhrase system, users register a multi-word phrase (with a minimum of two words) to authorise payments. A user can link multiple PayPhrases to the same underlying Amazon account, which is protected by a traditional password. Each PayPhrase is linked to a specific shipping address and payment card, allowing users to purchase items simply by typing in their phrase and a 4-digit PIN. Resistance to guessing attacks is expected to be provided both by the payphrase and the PIN.

Because no username is required, all PayPhrases must be unique. This prevents inferring the distribution of passphrases that humans will choose with no uniqueness restriction, which is common policy for password systems and necessary when passwords are used to protect private key files. This design choice allows us to study user selection of phrases simply by querying the publicly-accessible registration interface. As seen in Figure 1, the registration interface provides feedback to the user when attempting to select a phrase which has al-

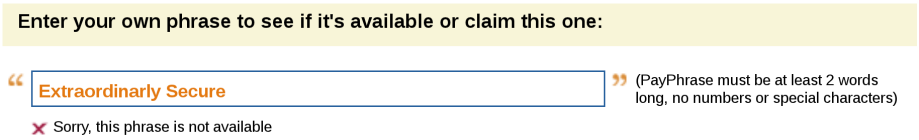


Fig. 1. The selection interface for passphrases deployed by Amazon.

ready been selected. We tested the registration status of over 100,000 possible passphrases using an automated script which queried this publicly-accessible interface While we found no evidence of rate-limiting, we limited our query rate to 1 Hz.

PayPhrases may only contain the space character and letters in the ASCII character set (the sets {a–z} and {A–Z}). No numbers, punctuation characters, or non-Latin letters are allowed. While PayPhrases must contain at least one space character at registration, spaces and capitalisation are ignored during verification. We will list all phrases we tested in a canonical lowercase form such as bases loaded.

3 Dictionary attack

Our first experiment was to simulate a dictionary attack by assembling a number of lists of phrases that English-speaking users might be expected to pick. We chose categories in part based on previous research on password guessing dictionaries [13,14], though this is an inherently subjective process.

Our first step was to query a large number of proper nouns of various categories, as summarised in Table 1. All of our proper nouns were taken from "top x" lists on Wikipedia,[2] except for lists of top movies and movie stars, which we took from the film-specific website IMDB.[3] We filtered the items in each list to comply with passphrase requirements, stripping punctuation and converting numbers and non-ASCII characters, as well as removing items which only contained one word. Overall, we tried more than 15,000 proper nouns.

We supplemented our list of proper nouns with a number of idiomatic phrases, summarised in Table 2. We obtained our sports phrases from Wikipedia, common English idiomatic phrases from the English teaching website English Language Learning Online,[4] and a list of the most popular slang expressions from the the online slang website Urban Dictionary.[5]

Our goal is to estimate the underlying probability of a user selecting an individual phrase from each category we identified. We first must approximate

[2] In some cases, the Wikipedia pages represented objectively collected lists, such as the largest cities in the world. In other cases, they were subjectively collected by Wikipedia editors as lists of notable items.

[3] `www.imdb.com`

[4] `www.usingenglish.com`

[5] `www.urbandictionary.com`

the total number of phrases selected. Based on a press release issued two months after our data collection experiments which claimed that now over a million users had registered a phrase, we take $N = 10^6$ as a rough estimate for the total number of phrases registered.

Given a set of n phrases, of which k were selected, we wish to approximate the probability p of any individual phrase in the list being selected. We make a key assumption that within each of our identified lists, all phrases have an equal probability of being selected. We further assume that each user who decides to register a phrase from our list picks randomly from the list. If the phrase the user picks is already selected, they then pick some other phrase not in the list.

Given that we've observed k selections from a list, the expected number of attempted selections k' is an instance of the *partial coupon collector's problem.* The first user attempting to select from our list will always succeed, the second user will succeed with probability $\frac{n-1}{n}$, the second with probability $\frac{n-2}{n}$, and so on. The expected number of attempts before the j^{th} phrase is selected is $\frac{n}{n-j}$ as a Bernoulli trial with $p_{\text{success}} = \frac{n-j}{n}$. Thus, the total number of attempts expected before k phrases are taken is:

$$\mathbf{E}[\#_{\text{attempts}}] = \prod_{j=1}^{k} \frac{n}{n-j} \tag{1}$$

Given that we observed k selections in a list of n from N total trials, we can then compute the maximum-likelihood probability of each item $\hat{p} = \frac{\mathbf{E}[\#_{\text{attempts}}]}{N \cdot n}$. In Table 1, $\hat{p}$ is listed for each category we tried.

3.1 Comparison to passwords

We estimate that our cumulative dictionary of 20,656 phrases covers the choices of about 1.13% of users. This level of security is equivalent to randomly-chosen strings of length $\lg\left(\frac{20{,}656}{0.0113}\right) \approx 20.8$ bits. For comparison, just 2 passwords (`123456` and `12345`) were chosen by 1.14% of users in the large dataset leaked from RockYou in 2009, equivalent to just 7.5 bits of security. Thus, passphrases appear to provide a significant boost in security over basic passwords against an attacker looking to compromise about 1% of accounts.

In another comparison, an optimal 20,656 word dictionary would cover 26.3% of passwords in the RockYou dataset. In an academic study, Klein manually assembled a dictionary in 1990 which covered over 9% of passwords with just 7,639 passwords [13]. These figures are equivalent to 16.3 or 16.4 bits of security, respectively. Thus, passphrases provide a security boost against attacks with small dictionaries by about 5 bits.

Our security estimates are slightly lower than those for mnemonic-phrase passwords by Kuo et al. [14], who found a 400,000 phrase dictionary which covered about 4% of choices, equivalent to 23.25 bits of security. Efficiency inherently declines with larger dictionaries, which partially explains this result. Additionally, Kuo et al. had to convert each phrase into a password. This can often be done in multiple ways, further making a dictionary attack less efficient.

word list	example	list size	success rate	$\hat{p}$
	arts			
musicians	three dog night	679	49.5%	0.0464%
albums	all killer no filler	446	56.5%	0.0372%
songs	with or without you	476	72.9%	0.0623%
movies	dead poets society	493	69.6%	0.0588%
movie stars	patrick swayze	2012	28.1%	0.0663%
books	heart of darkness	871	47.0%	0.0553%
plays	guys and dolls	75	70.7%	0.0093%
operas	la gioconda	254	17.3%	0.0048%
TV shows	arrested development	836	46.3%	0.0520%
fairy tales	the ugly duckling	813	13.3%	0.0116%
paintings	birth of venus	268	11.2%	0.0032%
brand names	procter and gamble	456	17.3%	0.0087%
total		7679	38.5%	0.4159%
	sports teams			
NHL	new jersey devils	30	83.3%	0.0056%
NFL	arizona cardinals	32	87.5%	0.0070%
NBA	sacramento kings	29	93.1%	0.0085%
MLB	boston red sox	30	90.0%	0.0074%
NCAA	arizona wildcats	126	56.3%	0.0105%
fantasy sports	legion of doom	121	71.1%	0.0151%
total		368	71.7%	0.0542%
	sports venues			
professional stadiums	soldier field	467	14.1%	0.0071%
collegiate stadiums	beaver stadium	123	12.2%	0.0016%
golf courses	shadow creek	97	6.2%	0.0006%
total		687	12.7%	0.0094%
	games			
board games	luck of the draw	219	28.8%	0.0074%
card games	pegs and jokers	322	27.6%	0.0104%
video games	counter strike	380	28.4%	0.0127%
total		921	28.2%	0.0306%
	comics			
print comics	kevin the bold	1029	29.5%	0.0361%
web comics	something positive	250	16.8%	0.0046%
superheros	ghost rider	488	45.3%	0.0295%
total		1767	32.1%	0.0701%
	place names			
city, state (USA)	plano texas	2705	33.8%	0.1117%
multi-word city (USA)	maple grove	820	79.0%	0.1283%
city, country	lisbon portugal	479	35.7%	0.0212%
multi-word city	ciudad juarez	55	69.1%	0.0066%
total		4059	43.7%	0.2677%
total		15481	38.1%	0.8479%

Table 1. Success rates of phrase dictionaries based on proper nouns.

word list	example	list size	success rate	$\hat{p}$
sports phrases	man of the match	778	26.1%	0.0235%
slang	sausage fest	1270	45.0%	0.0761%
idioms	up the creek	3127	43.6%	0.1789%
total		5175	41.3%	0.2785%

Table 2. Success rates of phrase dictionaries based on idiomatic phrases.

bigram type	example	list size	success rate
adverb-verb	probably keep	4999	5.0%
verb-adverb	send immediately	4999	1.9%
direct object-verb	name change	5000	1.2%
verb-direct object	spend money	5000	2.4%
verb-indirect object	go on holiday	4999	0.7%
nominal modifier-noun	operation room	4999	9.8%
subject-verb	nature explore	4999	1.3%

Table 3. Success rates of different classes of natural-language phrases taken from the British National Corpus [15].

4 Generated phrases

After exhausting simple dictionaries of the kind utilised in Section 3, a brute-force attack would require generating phrases according to a model of the underlying natural language. Given our online access to the Amazon oracle, we were unable to conduct a realistic brute-force search with millions of possible phrases. Instead, we conduct several experiments with randomly-generated phrases to evaluate linguistic tendencies in passphrase selection.

4.1 Phrases created using a syntactic parser

Our first linguistic question is, broadly, what type of syntactic constructions are most popular as passphrases? To address this, we evaluated random samples of naturally-occurring 2-word phrases of varying syntactic relation, extracted from the 100-million word British National Corpus [15] parsed by the Robust Accurate Statistical Parser [7,1]. All of the syntactic relations we tested were two words, except for indirect object relations where a preposition is required (e.g. pay in cash). We found vanishingly small numbers of longer phrases to be registered, preventing research on longer passphrases with this data source.

The list of grammatical relations we examined and the summary of results are presented in in Table 3. Of immediate interest, nominal modifier-noun phrases (e.g. bedtime story) were the most likely to be registered by nearly a factor of two. The next most popular list was adverbial-modifier verb relations (e.g. never leave), again twice as popular as any other list. This suggests that users prefer phrases which represent as a single object or a single action, rather than a verbal phrase containing an action and a subject or object.

bigram type	example	list size	success rate
adjective-noun	powerful form	10000	13.3%
noun-noun	island runner	10000	4.4%

Table 4. Success rates of bigrams taken from the Google n-gram corpus [6].

4.2 Phrases created using the Google n-gram corpus

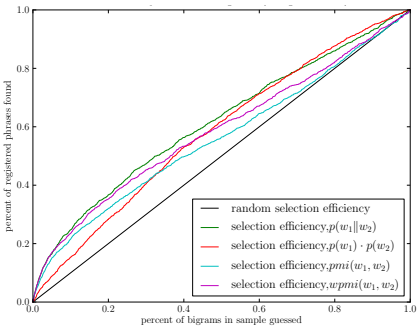
Our second linguistic question is, what factors predict how likely a given natural-language phrase is to be selected as a passphrase? We focus specifically on noun phrases using the much larger Google n-gram corpus which consists of over 10^{15} words of text harvested from the World Wide Web in 2006 [6]. Because this corpus contains counts for n-grams (sequences of n consecutive words) of only up to 5 words, sentence-level parsing is impossible. We instead relied on a much cruder classification of words as adjectives and nouns based on their most common part-of-speech tag in the RASP parsing of the BNC corpus [1].

We chose two random lists of 10,000 bigrams from the Google n-gram corpus, one consisting of adjective-noun bigrams and one of noun-noun bigrams. Basic statistics are given in Table 4. To evaluate how users may be selecting passphrases, we compared several potential models to rank each phrase in order selection probability. In Figure 2, we plot the percentage of registered phrases found against the percent of phrases guessed when proceeding in ranked order according to each model.

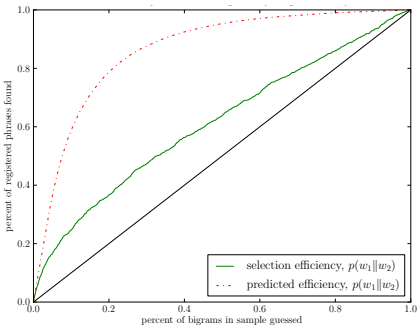
As a baseline, a random model considers users equally likely to pick any phrase from the list. This model produces a 45° degree diagonal line when plotted. We compare this to several other models:

- $p(w_1||w_2)$: bigrams are ranked by their overall probability. This simulates users generating passphrases exactly as pairs of words are generated in natural language.
- $p(w_1) \cdot p(w_2)$: bigrams are ranked by the product of the probabilities of each constituent word. This simulates users selecting each word in their phrase independently.
- $pmi(w_1, w_2)$: bigrams are ranked by the point-wise mutual information [8] of w_1 followed by w_2: $\lg \frac{p(w_1||w_2)}{p(w_1) \cdot p(w_2)}$. This simulates users having a tendency to pick words which are strongly associated with each other and hence occur together much more frequently than would be expected by random chance.
- $wpmi(w_1, w_2)$: bigrams are ranked by the point-wise mutual information of w_1 followed by w_2, multiplied by $p(w_1||w_2)$. This is a blended model.

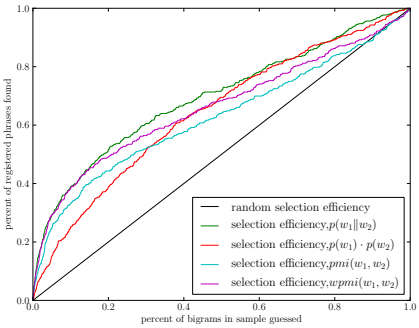
As seen in Figures 2a and 2c, the overall bigram probability is the best model for passphrase selection, though for the least-likely phrases, the independent probability model is just as accurate. Neither model based on pointwise mutual information provides additional predictive power. This leads us to conclude that users don't stray far from natural language patterns when choosing passphrases.



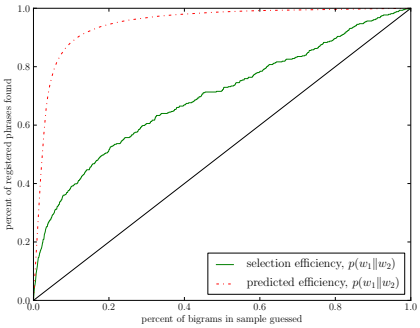
(a) Comparison of selection predictors (adjective-noun bigrams)



(b) Comparison of bigram probability to actual selection (adjective-noun bigrams)



(c) Comparison of selection predictors (noun-noun bigrams)

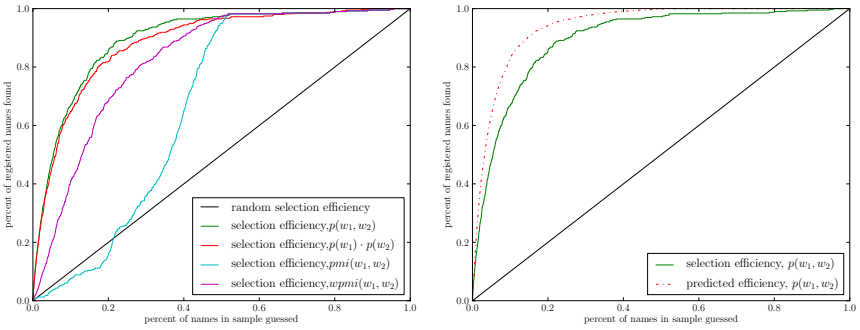


(d) Comparison of bigram probability to actual selection (noun-noun bigrams)

Fig. 2. The influence of different factors on the likelihood of individual bigrams being selected as passphrases. In Figures 2a and 2c, four different models are compared against a random-selection model: the overall bigram probability in the Google n-gram corpus, the product of the individual word probabilities, the pointwise mutual information of the bigram, and pointwise mutual information weighted by the overall bigram probability. In both cases, overall bigram probability is the best model. In Figures 2b and 2d, the expected efficiency of the overall bigram probability is compared to the observed effiency. In both cases, actual selection is considerably closer to random than predicted by the model.

However, this model is far from complete. In Figures 2b and 2d, we plot the expected efficiency if users perfectly followed the bigram probability model against our observed results. The large gap shows that users are considerably more random when choosing passphrases than when speaking naturally.

4.3 Phrases created from personal names



(a) Comparison of selection predictors for personal names

(b) Comparison of name probability to actual selection

Fig. 3. The influence of different factors on the likelihood of personal names (e.g. joseph bonneau) being selected as passphrases. The selection models are equivalent to those used defined in Section 4.2 and Figure 2.

A special class of phrases we identified are those based on a personal name, e.g. ekaterina shutova. Using 10,000 random names from a large corpus crawled from Facebook's public index of users in 2010 [3], we found 4% to be registered, a rate exceeding many of the types of natural language phrases as shown in Table 3. This is consistent with user preference for noun phrases.

We again tested several models for user selection of names as phrases as in Section 4.2, using the frequency of each name in the Facebook corpus as the overal "bigram probability" and the product of the frequencies of the first and last name from the Facebook corpus to simulate creating a random name, as plotted in Figure 3. In this case, these two models are nearly equivalent, as first and last names have relatively low mutual information compared to bigrams occurring in natural language; that is, being given the first name or last name of a person's name doesn't greatly help in guessing the other component. Still, as seen in Figure 3, guessing names in order of overall probability is the most effective model, with no indication that a name's point-wise mutual information influences user choice. As seen in Figure 3b, the model of users choosing a name for their passphrase at random according to the population-wide distribution of names produces very close results to our observed data.

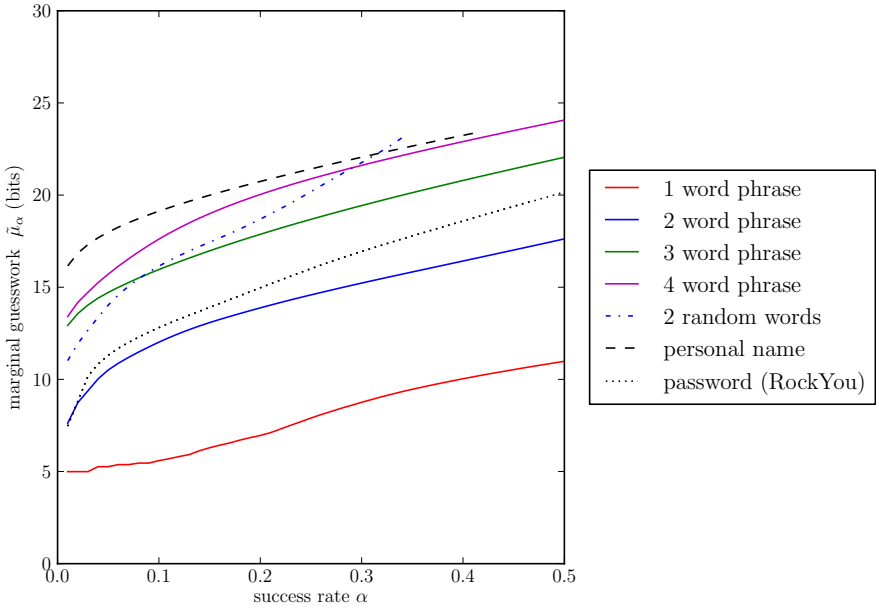


Fig. 4. The security provided by natural-language phrases of 1–4 words, based on estimated probabilities from the Google n-gram corpus. Also plotted is the difficulty of guessing a 2-word phrase if the words are selected independently, the difficulty of guessing a personal name based on the population distribution of names, and the difficulty of guessing a user-chosen password based on the leaked RockYou corpus.

4.4 Security implications

Given the evidence that user choice is partially predicted by the frequencies of phrases in natural language, it is natural to ask what security can be achieved if users in fact chose passphrases exactly in accordance with their distribution in natural language. We can examine this using the Google n-gram corpus to estimate of the probability distribution of multi-word phrases in English.

We use the *marginal guesswork* model to measure the guessing difficulty of a distribution [19,3,5]. The metric $\tilde{\mu}_\alpha$ measures the effective strength of a distribution in bits against an attacker desiring a α probability of guessing a user's passphrase correctly. It has been shown that no single metric can accurately measure guessing difficulty against attackers with different values of α [3]. Thus it is necessary to plot $\tilde{\mu}_\alpha$ across a range of values for α.

Figure 4 plots $\tilde{\mu}_\alpha$ for a phrases of 1–4 words, as well as randomly-chosen 2-word phrases, randomly-chosen names, and passwords. The results are somewhat discouraging for the passphrase concept, as 2-word phrases provide slightly less guessing resistance than existing text passwords. There is some gain from moving to 3-word phrases, but only a very small gain from 4-word phrases after that.

Given that we found users choose phrases more randomly than their natural language distribution, these findings should be considered a lower bound

for security. Many of the most common phrases in natural language are purely functional, such as as well as, and would be unlikely to be chosen as passphrases. Additionally, the Google n-grams corpus contains many artifacts of the web, with the most common 3-word phrase being all rights reserved and the most common 4-word phrase being property of their respective. Still, these findings suggest that multi-word phrases, if chosen naively according to natural language tendencies, are not as effective at mitigating guessing attacks as alternate choices, such as choosing 2 random words or choosing a personal name at random.

5 Concluding remarks

We consider our work preliminary due to the limitations of our dataset. In particular, without a full list of registered phrases, we can only test predicted selection strategies and there may be large classes of passphrases which we have not considered. Additionally, the unusual setup of the Amazon PayPhrase system may not encourage users to choose a difficult to guess password, as additional security is provided by a random PIN.

Our work suggests that multi-word passphrases have some promise as a means to improve security over traditional passwords. Even 2-word passphrases may be able to raise the security of the weakest selections from below 10 bits to over 20 bits which could be sufficient to make online attacks impractical. However, our results suggest that users aren't able to choose phrases made of completely random words, but are influenced by the probability of a phrase occurring in natural language. Examining the surprisingly weak distribution of phrases in natural language, we can conclude that even 4-word phrases probably provide less than 30 bits of security which is insufficient against offline attack. Our results are a caution against optimistic security estimates arising from Shannon's esimates of entropy [10] in place of of probabilities of whole phrases from modern corpora of natural language.

We recommend further collaboration between the security and linguistics research communities to explore what is possible in multi-word passphrases. In particular, user testing for longer phrases is necessary to determine the extent to which users will tend to choose passphrases with natural-language-like properties as more words are required and not resort to easier-to-remember patterns like repeated words, idioms, or well-known titles. We also suggest exploring random multi-word phrases in place of users-chosen ones, which our results suggest may allow improved guessing resistance with much shorter phrases.

Acknowledgements

Joseph Bonneau is supported by the Gates Cambridge Trust. Ekaterina Shutova is supported by the EU FP7 PANACEA project. We thank Diarmuid Ó Séaghdha for sharing his database of noun compounds.

References

1. Ø. Andersen, J. Nioche, E. J. Briscoe, and J. Carroll. The BNC Parsed with RASP4UIMA. In *Proceedings of LREC 2008*, 2008.
2. Gregory V. Bard. Spelling-error tolerant, order-independent pass-phrases via the damerau-levenshtein string-edit distance metric. In *Proceedings of the fifth Australasian symposium on ACSW frontiers - Volume 68*, ACSW '07, pages 117–124, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.
3. Joseph Bonneau, Mike Just, and Greg Matthews. What's in a name? Evaluating statistical attacks against personal knowledge questions. *FC '10: The Fourteenth International Conference on Financial Cryptography and Data Security*, 2010.
4. Joseph Bonneau and Sören Preibusch. The password thicket: technical and market failures in human authentication on the web. *WEIS '10: Proceedings of the Ninth Workshop on the Economics of Information Security*, June 2010.
5. Joseph Bonneau, Sören Preibusch, and Ross Anderson. Human selection and management of PINs. *FC '12: The Sixteenth International Conference on Financial Cryptography and Data Security (to appear)*, 2012.
6. Thorsten Brantz and Alex Franz. The Google Web 1T 5-gram corpus. Technical Report LDC2006T13, Linguistic Data Consortium, 2006.
7. Ted Briscoe, John Carroll, and Rebecca Watson. The second release of the RASP system. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, COLING-ACL '06, pages 77–80, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
8. Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Comput. Linguist.*, 16:22–29, March 1990.
9. Cormac Herley, Paul C. Oorschot, and Andrew S. Patrick. Passwords: If We're So Smart, Why Are We Still Using Them? pages 230–237, 2009.
10. Markus Jakobsson and Ruj Akavipat. Rethinking Passwords to Adapt to Constrained Keyboards. `www.fastword.me`, 2011.
11. Mark Keith, Benjamin Shao, and Paul John Steinbart. The usability of passphrases for authentication: An empirical field study. *International Journal of Human-Computer Studies*, 65(1):17–28, 2007. Information security in the knowledge economy.
12. Patrick Gage Kelley, Michelle L. Mazurek, Richard Shay, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Saranga Komanduri, and Serge Egelman. Of Passwords and People: Measuring the Effect of Password-Composition Policies. *The ACM CHI Conference on Human Factors in Computing Systems*, 2011.
13. Daniel Klein. Foiling the Cracker: A Survey of, and Improvements to, Password Security. In *Proceedings of the 2nd USENIX Security Workshop*, pages 5–14, 1990.
14. Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. Human Selection of Mnemonic Phrase-based Passwords. In *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*, pages 67–78, New York, NY, USA, 2006. ACM.
15. G. Leech. 100 million words of English: the British National Corpus. *Language Research*, 1993.
16. Andrew Mehler and Steven Skiena. Improving Usability Through Password-Corrective Hashing. In Fabio Crestani, Paolo Ferragina, and Mark Sanderson, editors, *String Processing and Information Retrieval*, volume 4209 of *Lecture Notes in Computer Science*, pages 193–204. Springer Berlin / Heidelberg, 2006.

17. Robert Morris and Ken Thompson. Password security: a case history. *Commun. ACM*, 22(11):594–597, 1979.
18. Adrian Perrig and Dawn Song. Hash Visualization: a New Technique to improve Real-World Security. In *In International Workshop on Cryptographic Techniques and E-Commerce*, pages 131–138, 1999.
19. John O. Pliam. On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks. In *Progress in Cryptology-INDOCRYPT 2000*, 2000.
20. Claude E. Shannon. Prediction and entropy of printed English. In *Bell System Technical Journal*, volume 30, pages 50–64, 1951.
21. Koichi Shimizu, Daisuke Suzuki, and Toyohiro Tsurumaru. High-Speed Search System for PGP Passphrases. In Matthew Franklin, Lucas Hui, and Duncan Wong, editors, *Cryptology and Network Security*, volume 5339 of *Lecture Notes in Computer Science*, pages 332–348. Springer Berlin / Heidelberg, 2008. 10.1007/978-3-540-89641-8_24.
22. Jeff Yan, Alan Blackwell, Ross Anderson, and Alasdair Grant. Password Memorability and Security: Empirical Results. *IEEE Security and Privacy Magazine*, 2(5):25, 2004.
23. Philip R. Zimmermann. *The official PGP user's guide*. MIT Press, Cambridge, MA, USA, 1995.