Isn’t that Fantabulous: Security, Linguistic and Usability Challenges of Pronounceable Tokens

Andrew M. White
amw@cs.unc.edu

Fabian Monrose
fabian@cs.unc.edu

Department of Computer Science

Katherine Shaw
kshaw@alumni.unc.edu

Elliott Moreton
moreton@email.unc.edu

Department of Linguistics

University of North Carolina at Chapel Hill

ABSTRACT

Over the past few decades, passwords as a means of user authentication have been consistently criticized by users and security analysts alike. However, password-based systems are ubiquitous and entrenched in modern society — users understand how to use them, system administrators are intimately familiar with their operation, and many robust frameworks exist to make deploying passwords simple. Unfortunately, much of the formal research on user authentication has focused on attempting to provide alternatives (e.g., biometrics) to password-based mechanisms (or belated analyses of users’ password choices), forcing administrators to use ad-hoc methods in attempts to improve security. This practice has lead to user frustration and inflated estimates of system security. We challenge common wisdom and re-examine whether *pronounceable* authentication strings might indeed offer a more reasonable alternative to traditional passwords. We argue that pronounceable authentication strings can lead to both improved system security and a decreased burden on users. To re-examine this potential, we explore questions related to how one might develop techniques for *rating* the pronounceability of word-like strings, and in doing so, enable us to quantify pronunciation difficulty. Armed with such an understanding, we posit new directions for *generating* usable passwords which are pronounceable and, we hope, memorable, hint-able and resistant to attack.

1. INTRODUCTION

Despite rampant criticism of passwords by users and security experts alike and an abundance of alternative proposals for user authentication, researchers have acknowledged that passwords are not likely to be replaced in the near future due to their ease of deployment and familiarity to users [13, 29]. Indeed, Bonneau et al. developed 25 different criteria intended to provide an objective viewpoint on the benefits and drawbacks of different user authentication schemes in terms of deployability, usability, and security; after evaluating passwords and 35 alternative schemes, Bonneau et al. ultimately concluded that no alternative scheme provides sufficient benefits to outweigh its detriments and overcome the current dominance of password-based systems [13].

NSPW 2014 Pre-Proceedings

We agree with that general sentiment, and believe it is time we refocus our attention on solutions that help improve the current state of affairs with password-based authentication. Over the past two decades, a number of policies for improving basic password systems have been suggested, but the most widely adopted of these suggestions is to increase the size of the space from which passwords are drawn (e.g., by enforcing the inclusion of numbers and special characters, requiring both upper and lower case letters, and increasing minimum password lengths). However, even for user-chosen secrets, these policies generally make passwords harder to remember and type, leading to user frustration and the habit of writing down and/or otherwise storing passwords [32, 68]. Worse yet, users generally fulfill these policy requirements in predictable ways, impairing, to a large extent, the security benefits these requirements are intended to provide [51, 63, 65]. Recently, Herley [28] rightfully argued that this behavior is, in fact, rational: users are often confronted with conflicting advice [37] and draconian password policies [32] for which they see marginal benefits [28]. Although alternatives to traditional textual passwords have been extensively explored and debated in the past [8] — particularly at NSPW [7, 23, 54, 57, 58] — we wish to stimulate further discussion and scientific exploration of methods which retain the benefits of textual passwords [13] while improving usability without sacrificing security.

In what follows, we argue that pronounceable authentication strings might offer a promising alternative to traditional passwords. In particular, we believe pronounceable authentication strings can lead to both improved system security and a decreased burden on users by providing memorable, hint-able passwords that are resistant to attack. We are intrigued by this direction because we believe early work on pronounceable passwords led to inconclusive results: while prior automated pronounceable password generators produced passwords which are both weak (due to their reliance on common English syllables) [38] and difficult to pronounce [17, 19], studies have also indicated that pronounceable passwords are easier for users to remember [52]. The latter finding suggests that the opportunity exists for leveraging linguistic expertise to develop techniques that provide memorable passwords through both user input and automated processes. One approach is to develop techniques for rating the pronounceability of word-like strings. In doing so, we believe we will be able to quantify pronunciation difficulty, which in turn, will allow us to proactively apply rigorous security analysis techniques to the space of pronounceable word-like strings to determine their suitability for use as secure passwords.

Our initial focus is on exploring the feasibility of automatically generating pronounceable passwords by forming lexical blends, sometimes known as *portmanteaus*, of two or more source words. By using pronounceable lexical blends, we believe these passwords will have a number of advantages over traditional systems, including *pronounceability*, *memorability*, *hintability*, and *resistance to attack*. We also envision scenarios where blends could be generated from one or more distinct semantic domains, which could be user-chosen. Generating blends from certain semantic domains naturally suggests one way in which we can provide *hintability*: the semantic domain(s) from which the source words are drawn can serve as hints (e.g., during the password reset process) to the user as to their password. Additionally, from a security perspective, system-generated lexical blends offer resistance to attacks as they will not be contained in any dictionary.

That said, we realize that in a proposition like ours, the security of the system depends on the space of possible blends; thus one must fully explore the security implications of that restriction. In particular, past research on analyzing the security of passwords has followed one of two approaches. The first is to consider only *system-generated* passwords, wherein the user is randomly assigned a password generated by the system in question. In this case, key properties to investigate include memorability and various negative externalities (such as password storing and user frustration) which result from passwords being assigned to users without any semantic context. The second approach is to consider empirically the choices of users, which we will refer to as *user-generated* passwords. Numerous other security-related questions abound, some of which we discuss herein.

Some of these questions might be addressed by an alternative type of password-generation, which we refer to as *user-influenced*, wherein a user can influence the formation of her password by, e.g., suggesting words to form part of a blend or suggesting semantic domains from which to draw source words, but the final password is ultimately system-generated. The hope is to aid in memorability by allowing the user to influence the formation of their (otherwise randomly generated) password. By allowing the system to ultimately assign the password, we may be able to maintain enough uncertainty in the distribution of possible passwords to avoid compromising the security of the system.

Finally, we believe that the ability to generate and rate pronounceable strings has value that extends beyond the realm of authentication. For instance, the automatic generation of domain names which are pronounceable and consistently spellable would be a boon for marketing. Alternatively, generating pronounceable domain names that are similar to a given *seed* string might aid in the proactive registration of potentially typosquatted domains. In addition, the generation and rating of (consistently) pronounceable and consistently spellable strings from restricted semantic domains may enable, e.g., the names of new drugs to be semantically relevant and distinctive while remaining easy for consumers and health care practicioners alike to remember, spell, and pronounce. Other potential applications include the replacement of tracking numbers and transaction IDs with sets of distinct word-like strings which improve, e.g., recognition over the phone, while retaining the error-correcting qualities of today's systems through metrics for measuring similarity in pronunciation. Finally, automatically generated yet pronounceable word-like strings might serve as an alternative to the 'short authentication strings' currently used to detect man-in-the-middle attacks on, e.g., VoIP calls following the ZRTP protocol [66].

2. RELATED WORK

Alternatives to memorizing passwords have been proposed, such as so-called *cognitive* [67] and *associative* [53] passwords. Cognitive passwords, also referred to as "personal knowledge questions", are now often used as password-reset mechanisms. However, cognitive passwords suffer from some of the same weaknesses as traditional passwords: in particular, they are often easily circumvented by targeted attacks using information gathered through acquaintances, phishing, social media and other publically available data sources [14, 46, 47]. In addition, personal knowledge questions are subject to similar statistical attacks as traditional passwords [12].

Associative passwords are based on the premise that, when *cued* with a certain word (or list of words), a given individual will respond with the same associated word or words each time [53]. However, Bunnell et al. [14] suggested that associated words with low "guessability"—i.e., those known to produce a wide range of responses among different people—were actually *harder* for users to recall than randomly generated passwords. However, Bunnell et al. also emphasized that the appropriate experimental conditions for testing the usability of associative passwords had not yet been identified, and that further research was necessary to determine the factors influencing both the memorability and usability of associative passwords [14].

Another approach is the use of pass*phrases*, consisting of multiple words rather than letters [45]. However, to date, there is little empirical evidence as to whether user-chosen passphrases offer any improvements over user-chosen passwords. Recent work suggests that users are unlikely to choose words with sufficient randomness to make passphrases secure against offline attacks [11]. In addition, a recent study by Shay et al., which compared system-assigned passphrases and passwords, found no significant benefit to either [52]. However, that study controlled for "guessability" by forcing relatively low entropies for the distributions from which passwords and passphrases were drawn, thus failing to take advantage of the much larger space of words than characters. Whether the memorability of passphrases (of equal length) scales with the size of the space from which the words are drawn remains an open question.

3. PRONOUNCEABLE PASSWORDS

The notion of "pronounceable" passwords was first explored by Gasser [20] and later standardized by NIST in 1993. This scheme, which we refer to as APG, operates by sampling at random from a set of base units (e.g., individual letter and certain pairs of letters), combining the sampled base units into valid syllables, then concatenating valid syllables to form a password. The probability distribution from which the base units are sampled is based on the unit frequencies in natural language; the concatenation at each step is governed by a complex set of rules also based on natural language [20].

Leonhard and Venkatakrishnan [38] proposed a pronounceable password generation scheme based on randomly selecting letters to form strings under two simple constraints (passwords may not begin or end with two consonants nor contain three consecutive consonants or vowels) intended to "ensure consistency with English spelling." An advantage of their scheme is that the resulting password space is easy to analyze relative to schemes based on occurrence frequencies.

Other pronounceable password generators operate by building passwords using phonemes (distinct sounds), *digraphs* (i.e., pairs of letters representing distinct sounds) and/or *trigraphs* as building blocks, n-gram character models, and character (e.g., vowel-vowel) substitutions [17]. Unfortunately, the pronounceable password generators discussed in this section (including APG), have been criticized for producing passwords which are difficult to pronounce [17, 19]. These criticisms, however, have been subjective in nature. In Section 4, we present our preliminary work on an objective metric for measuring pronounceability, and apply this metric to passwords generated by five different 'pronounceable' password generators.

3.1 Open Security Considerations

When the distribution of (potential) passwords (or a large enough sample thereof) is available, statistical methods can be used to analyze the security of passwords randomly drawn from such a distribution. Early analyses of password security focused on an attacker model in which the attacker targeted a specific user. In recent years, however, it is clear that attackers are more interested in compromising as many accounts as they can rather than targeting specific users. Accordingly, security analyses have evolved from techniques which fail to accurately model password guessing difficulty in multi-account scenarios to more sophisticated metrics.

Dictionary Attacks. When passwords, or computable hashes thereof, are available, analysis of the passwords has often been performed using similar tools to those used by attackers. These tools generally incorporate a number of password dictionaries, often termed "wordlists", and are capable of applying various transformations to the entries therein to provide variant passwords [37, 65]. The dictionaries used can be general or domain-specific: in particular, Kuo et al. explored the space of *mnemonic* passwords by collecting a dictionary of passwords formed by taking the first letter of each word in popular phrases (such as famous quotations or song lyrics). While their dictionary enabled them to crack a smaller proportion of mnemonic passwords than a standard dictionary could 'control' passwords, their results nonetheless suggest that such dictionaries may reduce the effectiveness of mnemonic-based password schemes [37].

Therefore, it seems prudent to investigate the effectiveness of building dictionaries of pronounceable word-like strings and employ these dictionaries in simulated attacks. Although similar to the attacks performed previously on conventional passwords [65], the transformations applied to the various entries in the dictionary would include not only character transformations but also *phonetic* or *syllabic* transformations on the pronunciation of the password to mimic the password-generation process. As part of our preliminary work, we have also developed methods for generating lexical neighbors of existing pronunciations, which will provide the basis for these sorts of attacks. One such method is to substitute phonemes or syllables in pronounceable strings with randomly-chosen replacements, then check the resulting string for phonotactic correctness and pronounceability. We welcome a discussion on other avenues for dictionary attacks that we may have overlooked.

Statistical Attacks. Ganesan and Davies proposed an attack on pronounceable password generators (specifically, a scheme implemented in Sandia's Kerberos V distribution and APG) which exploits the difference between the probability of a password belonging to a particular 'bucket' (e.g., starting with a particular unit) and the proportion of all the possible passwords which belong to that bucket. By concentrating on passwords in small buckets with high likelihoods, the attacker gains an advantage. Although this attack is valid for the Sandia scheme, which employs a first step in which one of 25 templates is chosen uniformly at random but the distribution of the number of possible passwords across templates is non-uniform [19], the proposed bucketing for the APG scheme does not produce the same effect using the base unit distribution given by Gasser [20].

However, statistical attacks on pronounceable passwords have not been examined in great detail, leaving many directions open which would be prudent to explore. In particular, we intend to explore the value of applying *α-work-factor* [44] and *α-guesswork* [10] metrics to the distributions generated by pronounceable passwords schemes in order to quantify their security.

3.2 Open Usability Considerations

We believe that using pronounceable word-like strings as authentication tokens provides usability benefits over traditional password schemes, particularly in terms of reducing the cognitive load on users. Specifically, we hypothesize that, with respect to traditional passwords, pronounceable word-like strings:

1. are easier for users to remember (*memorability*)[1]
2. are less frustrating for users (*acceptability*)
3. lead to fewer errors on token entry

[1] Interestingly, despite criticism on the grounds of pronounceability, Shay et al. [52] found that APG passwords were remembered slightly more often than passphrases.

Testing these hypotheses will require significant user studies for which rigorous and careful design is necessary. We believe the study of pronounceable passwords requires careful design decisions; many, but not all, of which have been explored and considered by this community in the past (e.g., [7, 23, 57, 58]). Specifically, a number of important questions remain open to our minds, particularly when considered in the specific context of using pronounceable word-like strings for authentication.

1. Subjective evaluations of performance on memory-related tasks correlate poorly with objective evaluations in many domains [48]. We intend to verify whether this phenomenon persists in the specific domain of pronounceable word-like strings; if so, then the question becomes: *are subjective evaluations more important in promoting the adoption of a system as a whole than objective measurements?* Separate assessment of subjective and objective criteria ameliorates, to an extent, the bias introduced by this phenomenon. That said, the role of *perception* in such studies has been overlooked and deserves greater exploration. Again, there are likely other factors that the research community has similarly disregarded, and we look forward to engaging discussions on factors that may limit the adoption of pronounceable tokens.
2. To what degree can studies in which participants are aware they are only using their password(s) for research provide ecological validity? As mentioned in Shay et al. [52], prior work has suggested that role-playing scenarios can influence users towards creating better passwords [34]. But do better passwords imply ecological validity?
3. The results of any study comparing a new authentication method with traditional passwords are inherently biased by the relative unfamiliarity of users with the new method [8]. Any studies on pronounceable passwords and particularly lexical blends would have to overcome similar bias, though perhaps less so than, e.g., graphical passwords. However, while the bias toward the familiar may be small due to the similarity between pronouceable and traditional passwords, the same similarity also complicates the analysis by introducing ambiguity as to the actual extent of any bias.
4. Allowing users a significant period of time in which to become accustomed to the new system highlights the issue of 'password interference' [16], where the necessity of maintaining multiple passwords in memory contributes to degraded performance. The extent (and direction) of the bias this introduces to studies of pronounceable passwords—which are both similar to traditional passwords and yet distinct—is another important variable.

Taken as a whole, we see the first two issues above as open questions which deserve attention from the community. In addition, we believe the contribution of the two confounding variables cannot be ignored, despite likely biasing results in opposing directions (a familiarity with textual passwords may induce a positive bias towards pronounceable passwords when compared with more radical alternatives while the effects of password interference may induce a negative bias). Since our proposed ideas differ in many ways from previous attempts to improve or replace traditional passwords, we solicit the community's feedback on appropriate study design decisions in the context of using pronounceable word-like strings as authentication mechanisms.

4. PRELIMINARY IDEAS

For the purposes of this thought-exercise, we consider a *word-like* string as an ordered pair (s, p) consisting of a spelling and a pronunciation. For example, the word-like string *slig* consists of the spelling *slig* and the pronunciation /slɪɡ/[2]. We consider a word-like string to be desirable based on three criteria, namely (1) **Wordlikeness**: It should look and sound normal for a word of that language, (2) **Consistent pronounceability**: Seeing the spelling s, all speakers should read it aloud in a similar way, as p, and (3) **Consistent spellability**: Hearing the pronunciation p, all speakers should infer the same spelling s. Wordlikeness, in terms of both lexical (text) and phonotactic (sound) similarity to known vocabulary, has been found to improve the memorability of a novel word [21, 22, 40, 41, 56]. Consistent pronounceability and spellability ensure that both the spoken and written forms of the string can be accurately learned from exposure to either of them. Examples of word-like strings satisfying or violating these criteria are shown in Table 1.

Rating Word-like Strings. Our current thinking for providing a rating metric is based on a simple scenario in which one person encounters an unfamiliar word in print and reads it aloud to another person, who then writes it down. The rating is a lower bound on the probability that what is written down by the listener is exactly what the speaker saw. In our preliminary proof-of-concept, we use a joint-sequence model of grapheme-to-phoneme and phoneme-to-grapheme conversion devised by Bisani and Ney [9]. A joint-sequence model is a model which maps sequences of symbols from one alphabet, e.g., *phonemes*, the distinct sounds which make up speech (such as the 'b' in 'bat'), to sequences of symbols from another alphabet, e.g., *graphemes*, the distinct characters which form written language. The joint sequences in question are represented by a sequence of *graphones*, where each graphone is a sequence of zero or more letters paired with a sequence of zero or more phonetic symbols.

The training input to this model is simply a lexicon of words, i.e., (s, p) pairs. By forming all possible decompositions of each word-pronunciation pair into graphones, the model learns a probability distribution over graphone sequences using standard n-gram techniques. Given a novel written word s, we modified the model of Bisani and Ney to return a set of pronunciations $\{p_1, \ldots, p_N\}$ together with associated conditional probabilities $\{\Pr(p_1 \mid s), \ldots, \Pr(p_N \mid s)\}$. Because of its symmetry, the joint-sequence model functions equally well in the other direction, accepting a novel pronunciation p and returning spellings with associated probabilities. In the exposition that follows, our model was trained on the Carnegie Mellon University Pronouncing Dictionary [62].

[2] In this paper, pronunciation is written using the International Phonetic Alphabet (IPA).

s	p	Wordlikeness	Consistent pronounceability	Consistent spellability
snib	/snɪb/	good	good	good
fip	/fɪp/	good	good	bad: *phip*
smough	/smoʊ/	good	bad: /smʌf/	bad: *smoe*
tlib	/tlɪb/	bad	good	good

Table 1: Examples illustrating the desirability criteria using word-like strings (of English). For those unfamiliar with IPA, /smoʊ/ rhymes with 'dough' while /smʌf/ rhymes with 'tough'.

For our application, the desirability of a spelled candidate s is defined by its *maximum path probability*:

$$R(s) = \max_{p} \Pr(p \mid s) \Pr(s \mid p).$$

The path probability $\Pr(p \mid s) \Pr(s \mid p)$ is the probability that s will be read as p, and p written as s, as in the scenario above, and is therefore a lower bound on the probability that what the listener writes down is exactly what the speaker saw.

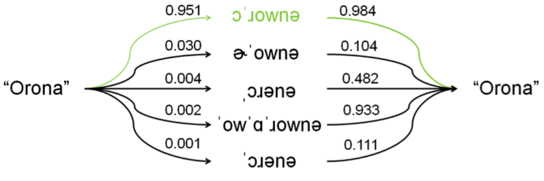

Figure 1: Possible paths for the string *Orona*.

Intuitively, graphones which occur frequently in the training set are assigned higher probability by the model than those which are rare or absent. Thus, the model provides the means to recognize and distinguish English-looking and English-sounding novel words from others that are less English-like. The most direct way to exploit this capability would be to measure the wordlikeness of (s, p) as the probability assigned by the model to (s, p). However, we used a simpler alternative, based on the observation that if a spelling s is atypical for English, there will be no conventional way to pronounce it and the model will be unable to parse the spelling into a sequence of graphones which is clearly more probable than others. Consequently, none of the paths from s through some p back to s will have high probability, and so $R(s)$ will be low. The maximum path probability metric thus incorporates wordlikeness as well as consistency of pronunciation and spelling.

For a concrete example consider the candidate *orona*, some of whose paths appear in Figure 1. The numbers associated with the top path, highlighted in green, mean that *Orona* has a .951 probability of being pronounced as [ɔˈɹownə], which in turn has a .984 probability of being written as *Orona*, according to the joint sequence model. Our preliminary rating metric provides us with the basic capability we need to rate the pronounceability of strings we generate. Obviously, this rating component can be used to filter the output that is shown to a user (in the case of system-generated authentication strings).

This preliminary metric also allows us to provide a quantitative analysis of previous work on pronounceable passwords. Our metric supports the previously mentioned criticism of 'pronounceable' password generators as producing difficult-to-pronounce passwords: as demonstrated in Figure 2, dictionary words (`cmudict`) score significantly higher using our metric than outputs from five different pronounceable password generators. The generators examined include two FIPS-181 implementations (`apg`[3] and `ogva`[4]), a trigraph-based algorithm (`gpw`[5]), a phoneme-based algorithm (`pwgen`[6]), and the character-based algorithm (`pronounce3`) proposed by Leonhard and Venkatakrishnan [38]. Perhaps unsurprisingly, the trigraph-based `gpw` (which considers the relative frequencies in English of three-letter sequences representing distinct sounds) produces passwords which are, in aggregate, scored slightly higher than those of the other generators (which are based on smaller distinct units).

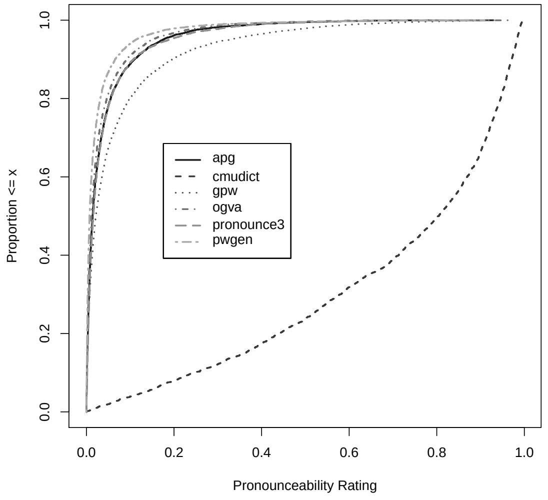

Figure 2: ECDF for ratings of 8-letter 'pronounceable' passwords from five different generators (`apg`, `gpw`, `pwgen`, `pronounce3`, and `ogva`; 5,000ea.) and 8-letter dictionary words (`cmudict`).

Estimating the Size of the Password Space. The pronounceable password generating algorithms mentioned above

[3] `http://www.adel.nursat.kz/apg/` (v2.2.3)
[4] `https://pypi.python.org/pypi/passogva/1.0` (v1.0)
[5] `http://www.multicians.org/thvv/gpw.html`
[6] `http://pwgen.sourceforge.net/` (v2.05)

are necessarily complex due to the complicated set of rules which governs spoken English. This complexity makes determining the resulting password spaces significantly more difficult than for traditional password schemes. An additional hurdle is that most of these algorithms are based on building passwords from component pieces that are spoken rather than written constructs. This necessitates a mapping from spoken form to written form before direct comparisons can be made.

To see why this mapping is important, consider a hypothetical generator which is based on stringing together n randomly selected syllables. The resulting written forms will vary in length significantly: in the CMU Pronouncing Dictionary [62], a one-syllable word's written form can be up to 9 characters in length. A two-syllable word ranges up to 14 characters, and a three-syllable word up to 16 characters. Therefore, a direct comparison with, e.g., 8-character passwords is impossible.

The complexity of previous algorithms lead Leonhard and Venkatakrishnan [38] to design their algorithm with ease of analysis as a major goal. In addition, Gasser [20] performed simulations to determine the size of the password space for `apg` passwords of 6, 8, and 10 characters. The resulting values are given in Table 2 along with estimates of the size of each space if the generated passwords are filtered to retain only the top 10% in terms of pronounceability (see Figure 2).

Generator	Length	Original	Pronounceable
`apg`	8	2^{30}	2^{27}
`pronounce3`	8	2^{31}	2^{28}
`apg`	10	2^{40}	2^{36}

Table 2: Approximate password space sizes for pronounceable password generators with fixed password lengths (in characters). The final column reduces the size of the space to 10% of the original based on the results in Figure 2 and the assumption that marginally pronounceable passwords are filtered out.

Generating Word-like Strings. As part of our preliminary explorations, we have also developed a naïve generator for pronounceable word-like strings. The generator was designed to produce English-like output by building syllables from the sub-syllabic constituents (*onsets*, *nuclei*, and *rimes*) found in the words of the CMU Pronouncing Dictionary [62]. Candidates are generated by concatenating one to three syllables, each of which was made by concatenating a random onset, nucleus, and coda. Each candidate, being a pronunciation, is converted to its most-probable spelling using the trained joint-sequence model described above. This yields candidates with a variety of lengths and sound shapes.

Even with this naïve method, we can explore methods for producing lexical neighbors of real words, i.e., word-like strings which are close in pronunciation to existing words. For example, to generate new word-like strings that sound like a given "seed" word, we can take the seed word's pronunciation and add, delete or replace phonemes to produce lexical neighbors. Candidates that are real words are removed from the set, as are those that cannot be parsed into legal syllables of English. The remaining candidates are converted to their most probable spellings and rated as before. Table 3 shows a handful of top-rated candidates for several seed words.

ANIMATION	ENERGY	SOFTWARE
animationer	tenergy	siftware
anvimation	yenergy	sunftware
animationed	venergy	sulftware
ganimation	henergy	loftware
panimation	ebnergy	seftware

Table 3: Example seeded-candidates for some sample words.

Unfortunately, these seeded word-like strings would fail miserably as passwords for the same reason that traditional passwords should not resemble dictionary words, i.e., most cracking dictionaries and methodologies include variants of dictionary words as highly likely candidates for password guessing. They may also be somewhat difficult to remember. Therefore, we turn our attention to more advanced methods for producing word-like strings based on generating *lexical blends*. It is our hope that in doing so, we will arrive at a rich source of pronounceable word-like strings which are not found in dictionaries nor too 'close' to a single existing dictionary word.

In what follows, we explore a more ambitious possibility: generating nonwords which simultaneously sound like two given seed words, such as *thoughtomotive* (resembling *thought* and *automotive*), or *evergy* (resembling *energy* and *ever*). Alternatively, to avoid words that sound similar to those on a given "blacklist", a filter stage can be added that compares the candidates to the blacklist and removes any that are closer than a user-specified distance.

4.1 Open Linguistic Considerations

Prounceable password generators must be able to generate nonsense words that humans would nonetheless find appealing; however, the properties that make such word-like strings appealing also represent opportunities for an adversary to narrow down and possibly rank-order the space of possible passwords. Therefore, we are concerned with the linguistic factors that affect the security of pronounceable passwords. In the present work, our focus is on using *lexical blends* as passwords, and therefore we focus on the linguistic factors which affect lexical blend formation.

Previous research has identified several factors that influence blend formation, such as the phonemic content of the source words, their length in syllables, how they are stressed, their internal syllabic structure, and their frequency of use [e.g., 2, 5, 6, 15, 25, 33, 59–61]. What (to our surprise) has not been systematically studied is how blend formation is affected by *meaning*, i.e., by the semantic domain in which the blend is formed, or by the intended meaning of the blend — *information known by the password holder but less likely to be available to an adversary.* Therefore, an important prerequisite to systematic use of lexical blends as passwords is an understanding of how meaningful context can affect blend formation.

One of our primary objectives is to design and implement an automatic blend generator (henceforth "blender"), which produces blends similar to those produced by humans, to aid with this understanding. Simply stated, the goal is to correctly predict the pronunciation of a blend from those of two given source words. In doing so, we hope to identify areas where there is variability in the pronunciation of a blend across different speakers. We believe it is important to focus on the variable blends because we hypothesize that two people blend the same word pair in different ways when they have different semantic definitions in mind. To simplify, we first consider candidates which begin like the first source word and end like the second (e.g., *br*unch from *br*eakfast + lunch). Even so, there are several important questions that must first be answered; we elaborate on some of them below.

Restricting the semantic domain. Blend formation is thought to be facilitated by *recoverability*, i.e., a blend is more acceptable to humans if the source words can be unambiguously inferred from it [5, 24]. For example, *education* + entertainment makes *edu*tainment, not *educate*ment, because entertainment is not recoverable from the latter — it could just as well be payment, achievement, etc. Recovery requires ruling out similar-sounding competitor words. Ordinary word recognition is faster and more accurate when context restricts the set of competitor words to associates (e.g., seeing *needle* facilitates recognition of *thread*) and relatives (e.g., *whale* helps with *dolphin*) of the target word [55]. We expect the same to be true of source-word recovery: blends will be more recoverable when they are processed in a restricted semantic domain. Hence, we expect meaningful context to make more blends possible, and to improve their memorability.

We hypothesize that users can form more blends when helped by meaningful context. Our preliminary results test this hypothesis, which builds upon established methodologies in the ordinary word-recognition literature [e.g., 26, 42]. In particular, our pilot study explores several questions, including but not limited to: (1) does semantic context make source words more recoverable from blends? If so, then people should be faster and more accurate at "solving" blends when context situates them in a restricted semantic domain. For instance, will tromboon be solved faster (as *trombone* + *bassoon*) when the context is the related word *orchestra* than when it is the unrelated *alcohol* or when no context is given? Use of the same blend with different contexts controls for the inherent plausibility of the blended concept, as well as word frequency and phonological factors. (2) Are more blends possible in a restricted semantic domain? If recoverability facilitates blending, and a restricted search space facilitates recoverability, then we expect more blends to be possible when context establishes a semantically-restricted search space.

To see why these questions matter, consider a real-life example: The cast of characters in a book, movie, TV series, etc. is a restricted semantic domain, and fans refer to characters collectively by blending their names. For instance, *Minerva* McGonagall and Severus Snape, from the Harry Potter series, are collectively *Minerv*erus or *Min*everus. Such blends are often non-recoverable to an outsider, though obvious to an insider. Sometimes one character's name is reduced to a single sound; e.g., *K*ock (*Kirk* plus Spock), *R*ico (*Rachel* plus Nico), or *S*aylee (*Simon* plus Kaylee).[7] Blends that are opaque and non-viable without context can become viable when the context is known. Context may thus increase the number of possible blends, while also making them memorable for those who know the context and baffling for those who do not.

Structural ambiguity and ambi-blendability. Arndt-Lappe and Plag [2] have shown that many source-word pairs can be blended in more than one way. We hypothesize that the choice among blend options is affected by the intended meaning of the blend, as a function of its morphological and syntactic structure. In our preliminary analyses, we found several systematic ways of generating such *ambi-blendable* source pairs. One is to use words which match phonologically at two "pivots", but diverge between them. For example, *flamingo* and mongoose match phonologically at the sound [m] and again at [ŋg]. The blend can switch words at either pivot, yielding two blend candidates — *fla*mongoose and *flamin*goose — which preserve different amounts of the two source words. Other methods include unstressed medial syllables that agree in onset but disagree in rime, like *Hufflepuff* + Gryffindor → *Huffle*dor or *Huff*inpuff, and onset clusters that could be switched at two points, like *blue* + green → *bl*een or *b*reen.

We suspect that the user's blending decision is affected by the intended meaning of the blend. Specifically, we hypothesize that the user's morphosyntactic parse of the source-word pair affects the choice of blend via a phenomenon (called *positional privilege*) which has been extensively studied in the ordinary-language phonology of the world's languages. Our recent work explores issues related to positional privilege in lexical blends [50]; our near-term goals include investigating how these issues might limit the space of pronounceable word-like strings that subjects create given two or more source words.

Estimating the Size of the Password Space. Similarly to simple pronounceable tokens, lexical blends present complications with respect to calculating the size of the password space. Previous work on lexical blends (e.g., [2, 5, 6, 15, 25, 33, 59–61]) has been from a bottom-up, inferential perspective, generally examining existing blends to determine the factors which result in the particular form that a blend of two source words takes and which is accepted into the language (e.g., why did blending 'breakfast' and 'lunch' result in 'brunch' rather than 'breakfunch'?). While these factors, along with the intended meaning of the blend, are important to consider (particularly from a usability standpoint), they tell us little about the space of potential blends. What is needed, instead, is a top-down analysis of how often two source words can be combined in such a way as to produce a blend which is *acceptable*, in terms of phonotactic and syllabic constraints, regardless of whether it's the *preferred* blend. Unfortunately, the complexity of spoken English makes this type of analysis difficult without extensive simulations.

[7] From `http://tvtropes.org/pmwiki/pmwiki.php/Main/PortmanteauCoupleName`, accessed November 23, 2012.

To see why extensive simulations are necessary, it is instructive to consider the many different forms which blends can take [49]. For instance, the combination of two source words can leave both words intact (as in *exam* + amnesia → *exam*nesia), leave one word intact but use only part of the other word (as in *decathlon* + athlete → *decath*lete), or leave neither word intact (as in *motor* + hotel → *mot*el). Furthermore, the two source words can share segments, i.e., overlap, in the blend (as in the examples in the previous sentence) or share nothing (as in *breakfast* + lunch → *br*unch). Finally, these examples cover only *linear* blend structures: other structures, such as embedded blends like *advertisement* + tease → *advert*ease*ment*, exist. Each of these possibilities must be investigated for each pair of source words and, for every possible pivot point or shared segment, the result tested against the phonotactic and syllabic constraints of spoken English.

The previous paragraph notwithstanding, we can make some rough estimates from our preliminary work on an automatic blend generator (our "blender") [49]. In order to ease the linguistic analyses, our preliminary blender is narrowly focused on one specific type of blend created from a particular subset of available word-pairs. As such, the result which follows is strictly a lower-bound on the number of possible blends and is likely to be extremely pessimistic. In particular, we used as our source of words the intersection of the CMUdict (as syllabified by Bartlett et al. [4]) and CELEX [1] lexicons (in order to eliminate many of the proper nouns in CMUdict). This resulted in a list of 36,216 words. We further restricted our attention to nouns consisting of exactly two syllables. From the possible pairs of words remaining, we selected those with a shared consonant between the nuclei of the two syllables to use as the blending point. This ensures that the blend would be phonotactically viable, resulting in 2,600,363 blends. We remind the reader that this figure is based only on pairs of disyllabic nouns blended at a shared internal consonant and is therefore an exceedingly loose lower-bound on the number of possible blends.

4.2 Possible Extensions & Associated Challenges

User Influence on Password Formation. Allowing user input into the password formation process fundamentally changes the way in which we must analyze password-based systems. In order to study these changes, we propose the use of both small, lab-based studies and larger scale crowd-sourced studies to generate pronounceable passwords under a variety of user-influenced conditions, as mentioned earlier. The passwords generated in these cases will then be subject to the same rigorous analyses as in the case of (fully) system-generated passwords. In addition, we fully anticipate that new avenues of attack will be unveiled during the course of this project, such as trends in user inputs which might provide an advantage to an attacker.

Prior work has analyzed biases in user-chosen passwords in both traditional and graphical password schemes [18, 65]. The work in Zhang et al. [65], for example, identified a number of distinct transforms which were common across many users, as well as identifying and exploiting the habits of particular users. For that reason, we believe it is also important to investigate whether our user-influenced password generation techniques lead users to form similar habits in password formation and/or generate passwords which are predictable from past passwords. Our prior work on 'phonetic edit distance' metrics [64] positions us perfectly to perform this analysis on pronounceable passwords by allowing us to measure the similarity of two passwords based on phonetic transformations as well as character transformations.

Unfortunately, even relatively naïve schemes for reducing the size of the potential pronounceable password space, such as rejecting any potential passwords which violate basic linguistic constraints, may substantially improve an attacker's chances. Therefore, a thorough analysis of pronounceable passwords and lexical blends is necessary and must include worst-case estimates. For instance, we intend to explore whether there exist hitherto unknown paradigms, e.g., dominant strategies in user formation of lexical blends, that an attacker might exploit. We also intend to assess the extent to which the use of pronounceable passwords and lexical blends impacts the effectiveness of shoulder-surfing. We welcome suggestions for other such dimensions to investigate.

Second-order Hints as a Password Reset Mechanism. While we suspect that the number of consistent blends for a given pair of source words will be too small to facilitate password-reset (or hinting) mechanisms if we simply show the user the two words and ask her to provide the "correct" blend, there are other possible ways to provide hints to a password. One method we intend to explore is to provide users with one of the source words. Others include (1) remind the user of the semantic domains of the source words, and (2) provide the user with hints whose connection to the source word is obscure to anyone who does not know the source word semantic domains. These methods are similar to *associative* passwords.

Consider, for example, the scenario where a user is asked for areas of interests during password setup. For pedagogical purposes, assume she specifies the semantic domains *genetics* and *knitting*. As output, the system-generated prounceable password "*homeo*socks" (from *homeobox* plus socks) is suggested, which she later accepts as her password. In the event that she forgets her password, the system could prompt her with *genetics* or *knitting*, which could jog her memory without giving the word away to an attacker. Less directly, it can prompt her with *regulate* or *feet*, which are associates of the target words only to someone who is already thinking in those domains. The user is likely to interpret the hints in the correct semantic domains (which she chose herself), allowing her to retrieve the hinted source word and hence the blend. For an adversary, the hints point in many more directions (e.g., *regulator* could lead to a fruitless search among scuba-diving terms).

On the other hand, ambi-blendability could result in blend passwords which are harder to hint, since the user might recall both source words, but still choose the wrong blend for them. Additionally, we found that 469,190 ambi-blendable word pairs of one very specific type could be made from a dictionary of 36,216 words suggests that ambi-blendable pairs (to say nothing of triplets or longer word strings) may be plentiful. Our results, however, also suggest precautions against this danger: by exploiting positional-privilege effects, hints could steer the user in the right direction with semantic context. For example, *Has a tail, steals things* is an appropriate hint for *baboo*ndit, whereas *Steals things with a tail* is appropriate for *ba*bandit—and not the other way around. Even an unsophisticated hint generator, such as that discussed previously, could make use of simple hint schemas (e.g., "X and Y" vs. "X of Y") to nudge the user one way or the other.

Natural Second Factors. There are at least two natural second factors for pronounceable-password-based authentication mechanisms. Since these passwords are inherently pronounceable, voice-based authentication suggests itself as a likely candidate. In addition, verbal input may be a viable alternative input mechanism (e.g., for mobile devices or over-the-phone exchanges) where text input is unavailable or inconvenient. The second natural candidate for two-factor authentication is *lip movement* as a biometric, especially because we can ensure that our system-generated passwords induce a threshold number of lip movements. Finally, prior work suggests that the combination of voice analysis and lip movement provides a reasonable biometric for speaker recognition [31]. It remains unclear to us which of these second factors might be the most fruitful to explore in the short term, and so we look forward to meaningful discussions and insightful feedback given that both textual passwords and biometrics have been hotly debated at NSPW in the past.

4.3 Preliminary Experiment Designs

Testing the hypotheses present in this work will require substantial experimentation, particularly in the form of user studies. In this section, we outline our thoughts on the major security-related experiments to perform. We stress that these are preliminary designs, and hope that the participants at the workshop will be able to aid us in detecting any oversights and suggesting new experiments and conditions to consider.

System-Assigned Tokens. Our first experiment would concern system-assigned authentication tokens, including traditional passwords, pass-phrases, pronounceable passwords, and lexical blends. We intend to build upon the work of Shay et al. [52], who performed a laudably thorough experiment and subsequent analysis, and their obvious and commendable commitment to full disclosure and reproducibility will no doubt aid us in our own work.

There are, however, aspects of Shay et al.'s experimental design which we disagree with and which we intend to rethink. In particular, Shay et al. used a fixed entropy value (of 30 bits) to determine the spaces from which the various authentication tokens they used were drawn. While this allows for straightforward comparison in terms of the security provided by these schemes, it mitigates the advantages of schemes which have a naturally higher entropy for the same cognitive requirements on the part of the user. For instance, from the point of view of a user, remembering a phrase consisting of three words each chosen at random from a dictionary of 200 words is no less difficult than remembering a similar phrase with words chosen from a dictionary of 10,000 words (provided the word list is not available to the user). Shay et al.'s study supports this contention in that the size of the dictionary (181, 401, or 1,024 words in their study) resulted in no significant difference in successful recall rate.

We suggest, therefore, that the different authentication schemes tested should not be artificially limited to a specific level of security, but rather allowed to assume the level of security provided in their 'natural' configuration. For instance, the median adult vocabulary size varies by age between 25,000 and 30,000 words.[8] Pass-phrase schemes with two, three, or four words chosen uniformly at random from a dictionary of size 25,000 have entropies of 29.2, 43.8, and 58.4 bits, respectively, the latter two a significant improvement over the 30 bits used as a baseline by Shay et al. without, we argue, a significant increase in cognitive load on the part of the user.

One might now argue that comparisons between schemes are no longer fair, since the security level of the system is no longer fixed. One approach to mitigating this problem is to compare each non-password scheme against a password scheme with the same level of security. Another approach would be to make relatively minor adjustments to the various schemes so as provide a small number of groups with roughly equivalent security. Finally, statistical measures, such as ANCOVA (analysis of covariance) [30], can control for the difference in security.

In addition to traditional passwords and pass-phrases, we intend to include system-assigned pronounceable tokens (and request pronounceability measurements from the users), lexical blends, and lexical blends of words chosen from restricted semantic domains. For the latter, conditions might include notifying the user of the semantic domain, using the semantic domain as a hint, and giving the user no notion of the domain at all. A comparison with pass-phrases under similar selection and presentation conditions is also warranted.

User-Influenced Tokens. Our second major experiment would focus on user-influenced tokens. Treatments for this experiment might include user-created lexical blends under various conditions: without any prompting (beyond an explanation of the general scheme), with a given semantic domain (possibly chosen by the user), with one word assigned by the system, and with the user choosing from a list of system-generated blends. For the sake of comparison, pass-phrase treatments might be included under similar conditions. Further conditions might include various second-order hint mechanisms, such as providing the user with a hint like the semantic domain, another word from the same semantic domain, or one of the source words. One primary goal would be to assess users' relative ability to recall tokens across different schemes. Additionally, users' preferences would be assessed. Finally, another primary goal would be to elicit an empirical distribution for each type of authentication token, to be used in analyses of the effective security levels such as have previously been performed for passwords (e.g., [10, 63]) and pass-phrases [11]. The results of these analyses could be used to inform any recommendations based on recall rates.

[8] `http://testyourvocab.com/blog/2013-05-08-Native-speakers-in-greater-detail`, accessed 2014-07-30.

5. CONCLUSION

With this work, we hope to call attention to both the promise and the challenges inherent in exploring the use of pronounceable word-like strings, and particularly lexical blends, as passwords. We outlined approaches to both rating the pronouceability of word-like strings and their automated generation, which we argue are essential tools for scientifically analyzing the feasibility of pronounceable passwords. Also central to that effort is an understanding of the linguistic properties of pronounceable passwords. Towards that end, we reported on our preliminary investigations into lexical blends, which we believe may serve as a viable pool of pronounceable, memorable and hintable passwords which are resistant to attack. Finally, we solicited the community's feedback on designing appropriate experiments to test our hypotheses, and highlighted a number of potential issues in experimental design which we believe are particularly relevant in the context of pronouceable passwords.

The questions and challenges discussed in the paper are by no mean the only issues worth exploring with regards to the applicability of user-influenced, yet system generated, pronounceable tokens. For instance, it is conceivable that allowing for error correction on the generated tokens might provide a substantial usability benefit. Indeed, this was already shown to be the case for passphrases, where users often misspell words when attempting to recall their passphrases [52]. In the case of pronounceable passwords, such error correction can be performed not only at the orthographic level but at the pronunciation level as well; e.g., 'piece' and 'peace' are pronounced identically but spelled differently. This sort of error correction may lead to increased usability for pronounceable passwords.

That said, incorporating error correction is a double-edged sword, as attackers may gain an advantage from the practice. Therefore, the limits of this advantage, from both phonetic and orthographic perspectives, must also be explored. We suspect that it may be possible to mitigate the attacker's advantage by incorporating password distinctiveness (phonetically, orthographically, or both) into the generation process (be it system generated or user-influenced). We may be able to use phonetic edit distance metrics [64] or lexical neighborhoods to determine whether isolating passwords in this way is feasible and to what extent this will reduce the space of pronounceable passwords.

It is also conceivable that some applications will call for novel words which resemble specific real words, while others may demand that the novel words *avoid* sounding like particular real words. Both desiderata require a metric of word similarity. One idea is to use the Levenshtein (string-edit) distance between pronunciations, which is defined as the minimum number of phoneme insertions, deletions, and substitutions required to turn one pronunciation into another [36]. In particular, the string-edit distance has been shown to influence perceived word similarity in psycholinguistic experiments [3, 27, 39] and in real-world confusions [35]. However, this glosses over the fact that edit distance does better if the different elementary operations are weighted differently, so that substituting similar-sounding phonemes contributes less distance than substituting very different ones. Nevertheless, this issue is also worth exploring.

6. ACKNOWLEDGMENTS

We are thankful to Steve Bellovin, Matt Bishop and Paul van Oorschot for insightful discussions and comments on an earlier draft of this manuscript.

References

[1] CELEX lexical database, 1995.

[2] S. Arndt-Lappe and I. Plag. Phonological variability in English blends. Handout from the Workshop on Data-Rich Approaches to English Morphology, Victoria University, Wellington, New Zealand, July 4–6 2012.

[3] T. Bailey and U. Hahn. Determinants of wordlikeness: Phonotactics or lexical neighborhoods. *Journal of Memory and Language*, 44(4):568–591, May 2001.

[4] S. Bartlett, G. Kondrak, and C. Cherry. On the syllabification of phonemes. In *Proceedings of Human Language Technologies*, 2009.

[5] O. Bat-El. Blend. In K. Brown, editor, *The Encyclopedia of Language and Linguistics*, volume 2, pages 66–70. Elsevier, Oxford, England, 2nd edition, 2006.

[6] O. Bat-El and E.-G. Cohen. Stress in English blends: a constraint-based approach. In *Cross-disciplinary perspectives on lexical blending.* Berlin, 2012.

[7] K. Bicakci and P. C. van Oorschot. A multi-word password proposal (gridword) and exploring questions about science in security research and usable security evaluation. In *NSPW*, 2011.

[8] R. Biddle, S. Chiasson, and P. Van Oorschot. Graphical passwords: Learning from the first twelve years. *ACM Comput. Surv.*, 44(4):19:1–19:41, Sept. 2012.

[9] M. Bisani and H. Ney. Joint-sequence models for grapheme-to-phoneme conversion. *Speech Communication*, 50(5):434–451, May 2008.

[10] J. Bonneau. The science of guessing: analyzing an anonymized corpus of 70 million passwords. In *IEEE Symposium on Security and Privacy*, 2012.

[11] J. Bonneau and E. Shutova. Linguistic Properties of Multi-word Passphrases. In *Workshop on Usable Security (USEC)*, pages 1–12, 2012.

[12] J. Bonneau, M. Just, and G. Matthews. What's in a name? Evaluating statistical attacks against personal knowledge questions. In *Financial Cryptography and Data Security*, volume 66, 2011.

[13] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *IEEE Symposium on Security and Privacy*, 2012.

[14] J. Bunnell, J. Podd, R. Henderson, R. Napier, and J. Kennedy-Moffat. Cognitive, associative and conventional passwords: Recall and guessing rates. *Computers & Security*, 16(7):629–641, 1997.

[15] G. Cannon. Blends in English word formation. *Linguistics*, 24:725–753, 1986.

[16] S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle. Multiple password interference in text passwords and click-based graphical passwords. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, 2009.

[17] H. Crawford and J. Aycock. Kwyjibo: automatic domain name generation. *Software: Practice and Experience*, 2008.

[18] D. Davis, F. Monrose, and M. K. Reiter. On user choice in graphical passwords schemes. In *USENIX Security Symposium*, 2004.

[19] R. Ganesan and C. Davies. A new attack on random pronounceable password generators. In *NIST National Computer Security Conference (NCSC)*, 1994.

[20] M. Gasser. A random word generator for pronouncable passwords. Technical report, MITRE Corporation, 1975.

[21] S. E. Gathercole, C. R. Frankish, S. J. Picering, and S. Peaker. Phonotactic influences on short-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(1):84–95, 1999.

[22] S. E. Gathercole, C. R. Frankish, S. J. Picering, and S. Peaker. Correction to Gathercole et al. (1999). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(3):562, 1999.

[23] M. Gibson, K. Renaud, M. Conrad, and C. Maple. Musipass: authenticating me softly with "my" song. In *NSPW*, 2009.

[24] S. T. Gries. Isn't that *fantabulous*? How similarity motivates intentional morphological blends in English. In M. Acard and S. Kemmer, editors, *Language, Culture, and Mind*, chapter 28, pages 415–428. CSLI Publications, Stanford, California, 2004.

[25] S. T. Gries. Shouldn't it be *breakfunch*? A quantitative analysis of blend structure in English. *Linguistics*, 42(3):639–667, 2004.

[26] F. Grosjean and U. Frauenfelder. A guide to spoken word recognition paradigms: Introduction. *Language and Cognitive Processes*, 11(6):553–558, 1996.

[27] U. Hahn and T. M. Bailey. What makes words sound similar? *Cognition*, 97:227–267, 2005.

[28] C. Herley. So long, and no thanks for the externalities: The rational rejection of security advice by users. In *New Security Paradigms Workshop (NSPW)*, 2009.

[29] C. Herley and P. Van Oorschot. A Research Agenda Acknowledging the Persistence of Passwords. *IEEE Security & Privacy Magazine*, 10(1):28–36, 2012.

[30] B. E. Huitema. *Analysis of Covariance (ANCOVA).* Sage Publications, Inc., 2007.

[31] M. Ichino, H. Sakano, and N. Komatsu. Multimodal biometrics of lip movements and voice using kernel fisher discriminant analysis. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision*, 2006.

[32] P. G. Inglesant and M. A. Sasse. The true cost of unusable password policies. In *SIGCHI Conference on Human Factors in Computing Systems*, 2010.

[33] M. H. Kelly. To "brunch" or to "brench": some aspects of blend structure. *Linguistics*, 36(3):579–590, 1998.

[34] S. Komanduri, R. Shay, P. G. Kelley, M. L. Mazurek, L. Bauer, N. Christin, L. F. Cranor, and S. Egelman. Of passwords and people: measuring the effect of password-composition policies. In *SIGCHI Conference on Human Factors in Computing Systems*, 2011.

[35] G. Kondrak and B. Dorr. Automatic identification of confusable drug names. *Artificial Intelligence in Medicine*, 36(1):29–42, 2006.

[36] J. B. Kruskal. An overview of sequence comparison: time warps, string edits, and macromolecules. *SIAM Review*, 25(2):201–237, 1983.

[37] C. Kuo, S. Romanosky, and L. F. Cranor. Human selection of mnemonic phrase-based passwords. In *Symposium on Usable Privacy and Security*, 2006.

[38] M. D. Leonhard and V. Venkatakrishnan. A comparative study of three random password generators. In *IEEE International Conference on Electro/Information Technology*, 2007.

[39] P. A. Luce. *Neighborhoods of words in the mental lexicon.* PhD thesis, Indiana University, 1986.

[40] S. Majerus, M. Van der Linden, L. Mulder, T. Meulmans, and F. Peters. Verbal short-term memory reflects the sublexical organization of the phonological language network: evidence from an incidental phonotactic learning paradigm. *Journal of Memory and Language*, 51:297–306, 2004.

[41] M. H. Messer, P. P. M. Leserman, J. Boom, and A. Y. Mayo. Phonotactic probability effect in nonword recall and its relationship with vocabulary in monolingual and bilingual preschoolers. *Journal of Experimental Child Psychology*, 105(4):306–323, 2010.

[42] J. H. Neely. Semantic priming effects in visual word recognition: a selective review of current findings and theories. In *Basic processes in reading: visual word recognition*, pages 264–336. 1991.

[43] NIST. Automated Password Generator (APG). Technical report, 1993.

[44] J. O. Pliam. On the Incomparability of Entropy and Marginal Guesswork in Brute-Force Attacks. In *Progress in Cryptology - INDOCRYPT*, volume 1977 of *Lecture Notes in Computer Science*, pages 67–79. 2000.

[45] S. N. Porter. A password extension for improved human factors. *Computers & Security*, 1(1):54–56, 1982.

[46] A. Rabkin. Personal knowledge questions for fallback authentication. In *Symposium on Usable Privacy and Security*, 2008.

[47] S. Schechter, A. B. Brush, and S. Egelman. It's No Secret. Measuring the Security and Reliability of Authentication via 'Secret' Questions. In *IEEE Symposium on Security and Privacy*, 2009.

[48] I. W. Schmidt, I. J. Berg, and B. G. Deelman. Relations between subjective evaluations of memory and objective memory performance. *Perceptual and Motor Skills*, 93(3), 2001.

[49] K. E. Shaw. Head faithfulness in lexical blends: A positional approach to blend formation, 2013.

[50] K. E. Shaw, A. M. White, E. Moreton, and F. Monrose. Emergent faithfulness to morphological and semantic heads in lexical blends. In J. Kingston, C. Moore-Cantwell, J. Pater, and R. Staubs, editors, *Proceedings of 2013 Meetings on Phonology*, Washington, DC. Linguistic Society of America.

[51] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor. Encountering stronger password requirements. In *Symposium on Usable Privacy and Security*, 2010.

[52] R. Shay, P. G. Kelley, S. Komanduri, M. L. Mazurek, B. Ur, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Symposium on Usable Privacy and Security*, 2012.

[53] S. L. Smith. Authenticating users by word association. *Computers & Security*, 6(6):464–470, 1987.

[54] A. Somayaji, D. Mould, and C. Brown. Towards narrative authentication: or, against boring authentication. In *NSPW '13: Proceedings of the 2013 Workshop on New Security Paradigms*. ACM Request Permissions, Dec. 2013.

[55] S. L. Thompson-Schill, K. J. Kurtz, and J. D. E. Gabrieli. Effects of semantic and associative relatedness on automatic priming. *Journal of Memory and Language*, 38:440–458, 1998.

[56] A. S. C. Thorn and C. R. Frankish. Long-term knowledge effects on serial recall of nonwords are not exclusively lexical. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 31(4): 729–735, 2005.

[57] J. Thorpe, P. C. van Oorschot, and A. Somayaji. Pass-thoughts: authenticating with our minds. In *NSPW*, 2005.

[58] J. Thorpe, A. Salehi-Abari, and R. Burden. Video-passwords: advertising while authenticating. In *NSPW*, 2012.

[59] R. Treiman. The structure of spoken syllables: evidence from novel word games. *Cognition*, 15:49–74, 1983.

[60] R. Treiman. The division between onsets and rimes in english syllables. *Journal of Memory and Language*, 25:476–491, 1986.

[61] R. Treiman, B. Kessler, S. Knewasser, R. Tincoff, and M. Bowman. English speakers' sensitivity to phonotactic patterns. In *Papers in Laboratory Phonology V: Acquisition and the Lexicon*, pages 269–282. 2000.

[62] R. Weide. The Carnegie Mellon pronouncing dictionary, 1998. URL `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[63] M. Weir, S. Aggarwal, M. 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 (CCS)*, 2010.

[64] A. M. White, K. Z. Snow, A. Matthews, and F. Monrose. Hookt on fon-iks: Phonotactic Reconstruction of Encrypted VoIP Conversations. In *IEEE Symposium on Security and Privacy*, 2011.

[65] Y. Zhang, F. Monrose, and M. K. Reiter. The security of modern password expiration. In *ACM Conference on Computer and Communications Security (ACM CCS)*, 2010.

[66] P. Zimmermann, A. Johnston, and J. Callas. ZRTP: Media Path Key Agreement for Unicast Secure RTP. RFC 6189, 2011. URL `http://www.ietf.org/rfc/rfc6189.txt`.

[67] M. Zviran and W. J. Haga. Cognitive passwords: The key to easy access control. *Computers & Security*, 9 (8):723–736, 1990.

[68] M. Zviran and W. J. Haga. Password security: an empirical study. *Management Information Systems*, 15(4):161–185, 1999.