A Comparative Study of Three Random Password Generators

Michael D. Leonhard
Department of Computer Science
University of Illinois at Chicago
uic@tamale.net

V.N. Venkatakrishnan
Department of Computer Science
University of Illinois at Chicago
venkat@cs.uic.edu
Member, IEEE

***Abstract*—This paper compares three random password generation schemes, describing and analyzing each. It also reports the results of a small study testing the quality of the passwords generated by the schemes. Qualities discussed include security, memorability, and user affinity. Improvements to the schemes and experiment are suggested.**

I. Introduction

Passwords are employed by nearly every multi-user computer application today. They are the most common user authentication method. Some systems allow each user to choose her own password while others create a random password for each user. In this paper, we focus on systems that employ random passwords and compare three schemes for generating such random passwords.

Random passwords are commonly used in e-commerce systems. Systems automatically generate passwords for users when they create website accounts or forget their passwords. Random passwords are also used in high security systems, such as military computers [1]. Generally, random passwords are used for one-time authentication and for applications where the user is expected to memorize the password and not write it down.

Random passwords have several benefits over user-chosen passwords. The main benefit is *security*. A random password generator creates passwords of specific entropy. This means that the password is chosen from a large set of potential passwords. In the average case, an attacker must search through half of the set to find a particular password. Using a password generator allows us to decide how much effort an attacker must expend to defeat the authentication system. On the other hand, when a person selects a password, there is no guarantee that the password comes from a large set. People often choose simple passwords that contain only a word and a number [2], [3]. Such passwords comprise a relatively small set and therefore are a smaller obstacle for attackers.

Some researchers have suggested instructing users to create mnemonic phrase-based passwords [4]. This advice is based on the assumption that such passwords will not appear in password cracking dictionaries and are therefore less vulnerable to attack. But Kuo, Romanosky, and Cranor showed that one can build a dictionary for such passwords [5]. Thus, for a sophisticated attacker, phrase-based passwords are about as easy to attack as word based passwords. They do not provide the security guarantee of randomly generated passwords.

The second benefit of random passwords is *confidentiality*. People often use the same password for multiple applications and websites [6]. When an attacker compromises one weak website, he can learn passwords for other websites and applications. When a person uses the same password on multiple accounts, she is setting up a fragile security dependency where a single breach leads to a total loss of security. Random assigned passwords increase security by forcing the person to use a unique password for the application. Of course, this security benefit is limited in situations where the person may adopt this application's random password for use on various other accounts.

The security benefits of random passwords are available with user-chosen passwords, if each user follows a suitable policy. The user can select a good password, which comes from a large password set. Additionally, she can use a unique password for each application and website account. But it is unrealistic to expect perfect compliance from users. For many websites, the user does not benefit from the presence of an account and password [6] and is therefore not motivated to follow a policy. On the other hand, an application can force unmotivated users into compliance by assigning a random password [7].

Aside from the benefits, there are usability concerns of random passwords. Random passwords are more difficult to remember than user-chosen passwords. When given the opportunity, the user will choose a password that has meaning to her [6]. She will have mental connections to the password to help remember it. An assigned password has no intrinsic meaning to the person receiving it. She will employ memorization strategies such as finding meaning in the random password and building mental connections. Unmotivated users loathe expending such effort. The difficulty of remembering a random password may drive the user to write down her password or simply stop using the website. Thus it is important to employ the best random password generation scheme to provide users with passwords that are easy to remember.

The purpose of this study is presented below, in Section II. Related work is outlined in Section III. Section IV describes the experimental procedure. We then describe and analyze the password generators, in Section V. The results of the experiment are in Section VI. Software for future studies is introduced in Section VII. Conclusions drawn from the experiment and suggestions for future work appear in Section VIII.

II. Purpose

This study considers three random password generation schemes, named AlphaNum, Diceware, and Pronounce3. The schemes were chosen because they are representative of different classes of schemes and are easy to analyze. The AlphaNum scheme constructs sequences of random characters. Diceware makes lists of words. Pronounce3 creates strings of syllables.

The purpose of this study is to find out which password generator produces the best quality passwords. The following metrics are used on the qualities of passwords generated in this study:

1) *Security*. The amount of entropy in each password
2) *Memorability*. How easily a normal user can remember the password
3) *Affinity*. How much the user likes the password

There are other characteristics of passwords that are not considered. The first concerns the length of the password. The schemes presented here can be easily extended to generate passwords of longer lengths and greater entropy. However, we considered the burden on users to remember long random passwords and chose scheme parameters that yield passwords of reasonable length.

Language is another characteristic of passwords. The schemes presented are designed for speakers of English, but may be modified to suit speakers of other languages.

III. Related Work

A similar study was performed by Bunnel, et. al. [8]. They compared user-generated passwords, randomly generated passwords, question-answer pairs, and word associations. Their participants correctly recalled 77% of user-generated passwords and 70% of randomly generated passwords. Their random password scheme was very simple. It concatenated a three-letter word, a numeral from 1 to 9, and a four-letter word. Although the security of the scheme is unsatisfactory, their study produced valuable experimental data. Their experiment served as a model for our study, presented here.

The US Department of Defense published guidelines for password management [1]. They present a technique for analyzing the security of passwords. We employ that technique in this study. They also suggest schemes that are very similar to the AlphaNum and Diceware schemes that we present here.

Two password managers [9], [10] were proposed along with claims about their usability. Chiasson, Oorschot, and Biddle performed studies [11] testing the usability of the software and found significant problems. Their study revealed several usability problems in these two password managers. Further, they stress the need for performing usability studies with real users. We followed their advice by performing a usability study of the schemes presented here.

IV. The Experiment

Our experiment consists of administering two questionnaires. The first contains a randomly generated password and tasks intended to help the participant to memorize it. The second questionnaire, given two weeks later, asks the participant to recall the password. The questionnaires are available for download from the project webpage [12]. The participants are undergraduate and graduate students taking a class on network security. The participants are likely to have a high understanding of security concepts and good password practices. We must keep in mind that the participants represent the upper echelon of the general user base when interpreting the results of the study. The questionnaire instructs each participant to treat the password as she would any other password. We also created a random password plugin [12] for the popular Wordpress [13] software. Future studies will use this software on a real webpage. This study, however, was done using paper and pencil.

For the first questionnaire, we ran each generator implementation to obtain 20 random passwords. This yielded 60 random passwords altogether. We then interleaved the order of the passwords. An AlphaNum password was first, then a Diceware password, then a Pronounce3, then another AlphaNum, and so on. We printed a questionnaire for each password. We handed out the questionnaires to participants by row, so people sitting next to each other would not receive the same type of password. Also, we distributed equal numbers of passwords of each type.

The first questionnaire contains instructions and a mockup webpage interface for a fictional website called Joe Maxwell Internet Auctions. The front side of this questionnaire is reproduced in the appendix as Fig.3. The participant is to role-play as a user of the website. Every view of the website contains the same logo and title.

The first "page" thanks the user for registering and displays her randomly generated password. Three subsequent "login screens" request the user to write her password in the password box and log in. If the user were to complete the questionnaire in a few moments, it is unlikely that she will remember the password later. We assume that retention is enhanced by lengthening the time for memorizing. Based on this assumption, we added meaningless time-consuming tasks between the login screens. The participants completed the first questionnaire in about 5 minutes.

The second questionnaire was administered two weeks after the first. All of these printed sheets were identical. The questionnaire instructs the participant to role-play logging into the website again. Three login screens are given, which are identical to those presented in the first questionnaire. Instructions ask the participant to try to remember her password and write it in the first login screen. Then, if she is uncertain of the password's correctness, she is to write other passwords that may be correct in the second and third login screens.

`dVysgZ`	`kraut gwen nagoya`	`ahzuphoste`
`a1LCLQ`	`voss terre snub`	`zuenacha`
`EDaL8p`	`plaid hey benz`	`vubagese`
`u1pbqY`	`isis uptake rca`	`zuwelopu`
`DbKrRZ`	`bryce aspire clone`	`agrofuxa`
`ED0uPw`	`doe slim dodo`	`fustuwchoi`
`tIG6QL`	`lv spiky coat`	`ezvedoxe`
`R7oBwn`	`fusty leper avon`	`yechnopee`
`YsM8Ht`	`portia toe trunk`	`ulciyolu`
`YpD1fD`	`lares ave ghent`	`epchigaxu`
(a) AlphaNum	(b) Diceware	(c) Pronounce3

Fig. 1. Outputs of the random password generators

The questionnaire then asks a few multiple choice questions, as shown in Fig.4 in the appendix. Next, there are two open-ended questions with space to write in responses. Finally, there is a space for the participant to write her email address if she wishes to receive a summary of the study results.

V. Password Generation Schemes

There are three random password generation schemes. For each scheme, we describe the technique used to generate passwords. This is followed by an analysis of the security of the passwords. Fig.1 contains ten passwords generated with each scheme. The source code of our implementations of these generators is available from the project webpage. [12]

A. AlphaNum *Generator*

This is the simplest generator. It creates a random password that is 6 characters long and may contain upper-case letters, lower-case letters, and numbers. The size of the alphabet is $26 + 26 + 10 = 62$. The generator chooses from this alphabet six times. The resulting password is the result of these six choices. There are 62 possibilities for the first character, 62 for the second, and so on. So the number of possible passwords is:

$$62 \times 62 \times 62 \times 62 \times 62 \times 62 = 62^6 = 5.68 \times 10^{10} = 2^{35.7}$$

This is the size of the password set. Mathematically, let P_A be the set of all possible passwords generated by this scheme. The size of the set is called the cardinality of P_A, denoted $|P_A|$. Because $|P_A| = 2^{35.7}$, we say that any password, p_A, chosen randomly from P_A, has 35.7 bits of entropy. We can use this measure of entropy to compare the strengths of various generators.

The purpose of this generator is to make passwords that are very short, yet contain enough entropy.

We implemented the AlphaNum generator in Python. The source code is available at [12]. Fig.1(a) is the output of the program running on Python 2.3.4 on Linux.

B. Diceware *Generator*

This generator produces random lists of words. It uses the idea that memorization requires one to form mental connections to the information being memorized. Every person knowing the meaning of a word has some kind of mental connection to it. By forming passwords with words, the person can take advantage of existing mental connections to make memorization easier. This kind of password is called a 'passphrase' by the US Department of Defense [1]. Reinhold provides a list of $7,776$ common words on his website [14]. He explains how to select passphrases using common six-sided dice, a technique he calls Diceware. We implemented the Diceware scheme in Python, using Reinhold's English word list. The source code and instructions for preparing the word list file are available at [12].

This generator independently chooses three words from the word list. Thus

$$|P_D| = 7776^3 = 4.70 \times 10^{11} = 2^{38.8}$$

This generator, Diceware, produces passwords with 38.8 bits of entropy. It is a little bit stronger than AlphaNum, which has 35.7 bits.

Fig.1(b) is the output of the program running on Python 2.3.4 on Linux. As we can see, some of the words are rather obscure. Passwords may contain words that users do not know. For example, the authors were unaware of the meanings of the words 'portia', 'lares', and 'ghent.' Reinhold's technique utilizes six-sided dice and requires $7,776$ words. But our program may use a word list of any size. For future work, less common words may be removed from the word list.

C. Pronounce3 *Generator*

The Pronounce3 scheme produces passwords that are pronounceable in English. The objective of this scheme is to utilize the speech facilities of the user's mind to assist in remembering the password.

Ganesan and Davies describe a major flaw in pronounceable password generators [15]. The generators choose syllables based on their frequency in English writing, using complex rules to achieve pronounceability. The result is that some passwords are more likely to be chosen than others. Ganesan and Davies show how this lack of uniform probability ruins the security of the generators.

The Pronounce3 scheme does not have the flaw described by Ganesan and Davies. It takes a simple approach to password construction resulting in uniform entropy for all passwords in the password space. We can easily analyze the security of the generator.

The Pronounce3 generator composes passwords of consonant and vowel elements. There are five vowels:

```
a, e, i, o, u
```

There are twenty two consonants:

```
b, c, ch, d, f, g, h, j, k, l, m,
n, p, ph, r, s, st, v, w, x, y, z
```

To ensure consistency with English spelling, we restrict password composition with two rules:

1) No password may begin or end with two consonants.

2) The password may not contain three consecutive consonants or three consecutive vowels.

Given a certain number of vowels and consonants, there are various orderings that satisfy the two restrictions. The scheme represents an ordering with a template string. A template is a string of α and β symbols, where α represents a vowel and β, a consonant. The set of templates with v vowels and c consonants is denoted $T_{v,c}$. The set of templates using 4 vowels and 4 consonants contains thirteen elements:

$$T_{4,4} = \left\{ \begin{array}{l} \alpha\alpha\beta\beta\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\beta\alpha\beta\alpha, \\ \alpha\beta\beta\alpha\alpha\beta\beta\alpha, \alpha\beta\beta\alpha\beta\alpha\beta\alpha, \alpha\beta\beta\alpha\beta\beta\alpha\alpha, \\ \beta\alpha\alpha\beta\alpha\beta\beta\alpha, \beta\alpha\alpha\beta\beta\alpha\beta\alpha, \beta\alpha\beta\alpha\alpha\beta\beta\alpha, \\ \beta\alpha\beta\alpha\beta\alpha\beta\alpha, \beta\alpha\beta\alpha\beta\beta\alpha\alpha, \beta\alpha\beta\beta\alpha\alpha\beta\alpha, \\ \beta\alpha\beta\beta\alpha\beta\alpha\alpha \end{array} \right\}$$

Given sets of templates, vowels, and consonants, password generation begins by randomly choosing one of the templates. The scheme then iterates through the template. When an α is encountered, it randomly chooses a vowel and appends it to the password. Each vowel is equally likely to be chosen. Similarly, for a β, it appends a random consonant.

Let us denote the set of all passwords generated by the scheme as $P_{v,c}$ where v is the number of vowels and c is the number of consonants. Since $T_{v,c}$ is the set of valid templates that contain v vowels and c consonants, it should be plain that

$$|P_{v,c}| = |T_{v,c}| \times 5^v \times 22^c$$

For this study, we use the PRONOUNCE3 scheme to generate passwords containing 4 vowels and 4 consonants. We implemented this in Python. The program source code is available at [12]. Fig.1(c) is the output of the program running on Python 2.3.4 on Linux. The generator chooses passwords from the set $P_{4,4}$.

$$|P_{4,4}| = 13 \times 5^4 \times 22^4 = 1.90 \times 10^9 = 2^{30.8}$$

The generator's 30.8 bits of entropy are less than ALPHANUM's 35.7 bits and DICEWARE's 38.8 bits. We considered several ways to increase the entropy of this generator. One way is to introduce more templates. This requires different numbers of vowels and consonants. Table 1 lists the eighteen non-empty password sets whose passwords have length eight or less. Note that this length is the number of vowel and consonant elements. Some elements, such as 'ch', contain two characters. Passwords containing such elements are longer than eight characters.

From Table I, we can see that $P_{4,4}$ is the largest set. As shown previously, using only $P_{4,4}$ yields passwords with 30.8 bits of entropy. Consider modifying the scheme to choose passwords from $P_{4,4} \bigcup P_{3,5}$.

$$|P_{4,4}| + |P_{3,5}| = 1.90 \times 10^9 + 6.44 \times 10^8 = 2.54 \times 10^9 = 2^{31.2}$$

By adding $P_{3,5}$, we gain a negligible 0.4 bits of entropy. The question is whether we can do better if we include all of the valid password sets. Consider the following equation:

$$|P_{1,0}| + |P_{1,1}| + |P_{2,0}| + \ldots + |P_{5,3}| + |P_{6,2}| = 3.16 \times 10^9 = 2^{31.5}$$

This is hardly any better than using only $P_{4,4}$. In fact, by using all of the sets, we gain only 0.76 bits of entropy. Clearly, to achieve higher entropy, the scheme must allow some templates that contain nine elements. That is an area for future study.

Another area to investigate is the addition of upper-case letters. By allowing the first letter to be either upper-case or lower-case, we gain one bit of entropy. Various other capitalization schemes deserve investigation, too. Another promising modification is the addition of symbols such as the hyphen, period, asterisk, etc.

TABLE II
PARTICIPANTS COMPLETING BOTH QUESTIONNAIRES AND RECALLING THEIR PASSWORDS

	ALPHANUM	DICEWARE	PRONOUNCE3
Total Participants	6	7	6
Recalled Password	1	2	1

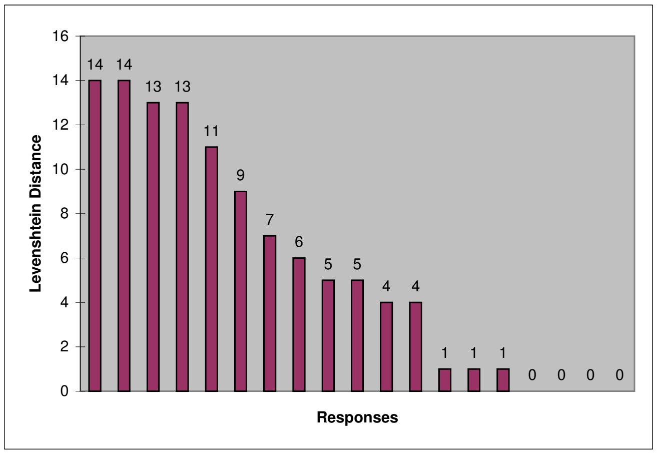


Fig. 2. Levenshtein distance of recalled passwords to assigned passwords

VI. RESULTS

The experiment used the passwords in Fig.1. Twenty nine people participated in the first part of the experiment, receiving a password on the first questionnaire. Nineteen of those people completed the second part of the experiment, properly filling out the second questionnaire. Table II lists the distribution of passwords to the participants and their recollection rates.

No participant wrote an incorrect password in the first login box and subsequently wrote a correct password in the second or third boxes. If the first response was incorrect, so were the others. Some participants recalled their passwords but were mistaken in one letter. Others omitted a letter.

The Levenshtein Distance is the number of edits required to transform their first response into the correct password. It represents how close the user's response was to the correct response. See Fig.2.

Password affinity was queried with the question, "How do you like your password?" After converting the responses to numerical values, we can compare the responses for the various schemes. Here is the coding: 'hate it' = 0, 'don't like it' = 1, 'ok' = 2, 'like it' = 3, 'love it' = 4. Table III lists the

TABLE I
ALL EIGHTEEN NON-EMPTY PASSWORD SETS AND THEIR PROPERTIES

v	c	$\lvert P_{v,c}\rvert$	$\lvert T_{v,c}\rvert$	$T_{v,c}$
1	0	5.00×10^0	1	$T_{1,0} = \{\alpha\}$
1	1	1.10×10^2	1	$T_{1,1} = \{\beta\alpha\}$
2	0	2.50×10^1	1	$T_{2,0} = \{\alpha\alpha\}$
2	1	1.10×10^3	2	$T_{2,1} = \{\alpha\beta\alpha, \beta\alpha\alpha\}$
2	2	2.42×10^4	2	$T_{2,2} = \{\alpha\beta\beta\alpha, \beta\alpha\beta\alpha\}$
2	3	2.66×10^5	1	$T_{2,3} = \{\beta\alpha\beta\beta\alpha\}$
3	1	5.50×10^3	2	$T_{3,1} = \{\alpha\alpha\beta\alpha, \alpha\beta\alpha\alpha\}$
3	2	3.03×10^5	5	$T_{3,2} = \{\alpha\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\alpha, \alpha\beta\beta\alpha\alpha, \beta\alpha\alpha\beta\alpha, \beta\alpha\beta\alpha\alpha\}$
3	3	6.66×10^6	5	$T_{3,3} = \{\alpha\beta\alpha\beta\beta\alpha, \alpha\beta\beta\alpha\beta\alpha, \beta\alpha\alpha\beta\beta\alpha, \beta\alpha\beta\alpha\beta\alpha, \beta\alpha\beta\beta\alpha\alpha\}$
3	4	8.78×10^7	3	$T_{3,4} = \{\alpha\beta\beta\alpha\beta\beta\alpha, \beta\alpha\beta\alpha\beta\beta\alpha, \beta\alpha\beta\beta\alpha\beta\alpha\}$
3	5	6.44×10^8	1	$T_{3,5} = \{\beta\alpha\beta\beta\alpha\beta\beta\alpha\}$
4	1	1.38×10^4	1	$T_{4,1} = \{\alpha\alpha\beta\alpha\alpha\}$
4	2	1.51×10^6	5	$T_{4,2} = \{\alpha\alpha\beta\alpha\beta\alpha, \alpha\alpha\beta\beta\alpha\alpha, \alpha\beta\alpha\alpha\beta\alpha, \alpha\beta\alpha\beta\alpha\alpha, \beta\alpha\alpha\beta\alpha\alpha\}$
4	3	7.32×10^7	11	$T_{4,3} = \left\{ \begin{array}{l} \alpha\alpha\beta\alpha\beta\beta\alpha, \alpha\alpha\beta\beta\alpha\beta\alpha, \alpha\beta\alpha\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\alpha\beta\alpha, \alpha\beta\alpha\beta\beta\alpha\alpha, \alpha\beta\beta\alpha\alpha\beta\alpha, \\ \alpha\beta\beta\alpha\beta\alpha\alpha, \beta\alpha\alpha\beta\alpha\beta\alpha, \beta\alpha\alpha\beta\beta\alpha\alpha, \beta\alpha\beta\alpha\alpha\beta\alpha, \beta\alpha\beta\alpha\beta\alpha\alpha \end{array} \right\}$
4	4	1.90×10^9	13	$T_{4,4} = \left\{ \begin{array}{l} \alpha\alpha\beta\beta\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\alpha\beta\beta\alpha, \alpha\beta\alpha\beta\beta\alpha\beta\alpha, \alpha\beta\beta\alpha\alpha\beta\beta\alpha, \alpha\beta\beta\alpha\beta\alpha\beta\alpha, \alpha\beta\beta\alpha\beta\beta\alpha\alpha, \beta\alpha\alpha\beta\alpha\beta\beta\alpha, \\ \beta\alpha\alpha\beta\beta\alpha\beta\alpha, \beta\alpha\beta\alpha\alpha\beta\beta\alpha, \beta\alpha\beta\alpha\beta\alpha\beta\alpha, \beta\alpha\beta\alpha\beta\beta\alpha\alpha, \beta\alpha\beta\beta\alpha\alpha\beta\alpha, \beta\alpha\beta\beta\alpha\beta\alpha\alpha \end{array} \right\}$
5	2	4.54×10^6	3	$T_{5,2} = \{\alpha\alpha\beta\alpha\alpha\beta\alpha, \alpha\alpha\beta\alpha\beta\alpha\alpha, \alpha\beta\alpha\alpha\beta\alpha\alpha\}$
5	3	4.33×10^8	13	$T_{5,3} = \left\{ \begin{array}{l} \alpha\alpha\beta\alpha\alpha\beta\beta\alpha, \alpha\alpha\beta\alpha\beta\alpha\beta\alpha, \alpha\alpha\beta\alpha\beta\beta\alpha\alpha, \alpha\alpha\beta\beta\alpha\alpha\beta\alpha, \alpha\alpha\beta\beta\alpha\beta\alpha\alpha, \alpha\beta\alpha\alpha\beta\alpha\beta\alpha, \alpha\beta\alpha\alpha\beta\beta\alpha\alpha, \\ \alpha\beta\alpha\beta\alpha\alpha\beta\alpha, \alpha\beta\alpha\beta\alpha\beta\alpha\alpha, \alpha\beta\beta\alpha\alpha\beta\alpha\alpha, \beta\alpha\alpha\beta\alpha\alpha\beta\alpha, \beta\alpha\alpha\beta\alpha\beta\alpha\alpha, \beta\alpha\beta\alpha\alpha\beta\alpha\alpha \end{array} \right\}$
6	2	7.56×10^6	1	$T_{6,2} = \{\alpha\alpha\beta\alpha\alpha\beta\alpha\alpha\}$

TABLE III
AVERAGES OF RESPONSES TO THE QUESTION "HOW DO YOU LIKE YOUR PASSWORD?"

	Mean
All Schemes	1.73
ALPHANUM	1.67
DICEWARE	1.71
PRONOUNCE3	1.83

results of this analysis. The numbers indicate that participants liked the passwords from the PRONOUNCE3 scheme a little bit more than the other schemes. Because of the small sample size, this difference is probably within the margin of error.

Responses to the open-ended questions at the end of the second questionnaire were enlightening. Four participants reported using rote memorization. One participant remarked, "I tried to recollect it often (of course, not that frequently)."

Six participants reported using mnemonic techniques to associate meaning with portions of their passwords. One wrote, "It was very hard to remember, because there were no meaningful words in them that could be remembered."

Four participants indicated that repeated use would have helped them to remember their passwords. One participant wrote, "I don't remember anything well. Only repetition over many days will I remember it."

VII. WORDPRESS INTEGRATION

WordPress [13] is a popular blogging platform. The second author uses WordPress for his class web pages. Each student in a course is given an account with permissions to post comments on the blog. The course blog facilitates discussion of course material and assignments. The authors have created a random password plugin for WordPress. The plugin replaces the password selection functionality of WordPress with random password generation and assignment. The plugin is available for download at [12]. Future studies will use this plugin on the course webpages.

VIII. CONCLUSION & FUTURE WORK

The results of the study show that there is room for improvement in random password generators. From the security analysis, we learned that the generators may be adjusted to yield longer or shorter passwords.

The study also shows that people have difficulty remembering random passwords, even those in the upper echelon of the general user base. Random passwords may be very useful when used with security tools [9], [10] that reduce the burden on the user's memory. Such tools allow the user to securely replace many infrequently used passwords with one that is frequently used and therefore less likely to be forgotten.

The open-ended question responses direct us to ways we can improve the schemes. It might be helpful to generate mnemonic aids for ALPHANUM passwords. The DICEWARE scheme may be improved by removing obscure words from the word list. The PRONOUNCE3 scheme could gain entropy by the addition of capital letters and punctuation.

Another suggestion is to train each user to remember her password. The software could teach mnemonic techniques and provide exercises and quizzes.

Participants' performance also points out some areas for improvement. Four participants recalled their passwords perfectly. Additionally, three participants made only one mistake in their passwords. The schemes may be improved to prevent these minor faults in recollection. For example, one participant incorrectly remembered 'yechnopee' as 'yecknopee'. The person may have memorized the 'ech' sound as 'eck', resulting in an error. Removing the 'ch' consonant element could prevent future users from making this mistake. Similarly, ALPHANUM may be improved by eliminating easily confused pairs such as 'n' and 'm'.

A future experiment will evaluate these schemes implemented in a Wordpress class blog. The participants will log into the website regularly to download homework assignments and study aids. Each person would use her assigned password regularly. The website will record events such as successful logins, failed logins, password reminders, etc. This information will form the basis of a better comparison of the password generation schemes.

ACKNOWLEDGMENTS

The authors would like to thank the students of the Fall 2006 undergraduate computer security class at UIC for participating in the study that is discussed in this paper.

REFERENCES

[1] R. L. Brotman, *Department of Defense Password Management Guideline*, CSC-STD-002-85, Department of Defense Computer Security Center, 1985.

[2] O. Fredstie. (2006, November) End users attitudes and behaviours towards password management: Survey report. Dept. of Information Science, University of Otago, New Zealand. [Online]. Available: http://www.fredstie.com/thesis/survey/survey_report.pdf

[3] B. Schneier. (2006, December) Myspace passwords aren't so dumb. Wired News. [Online]. Available: http://www.wired.com/news/culture/0,72300-0.html

[4] D. V. Klein, ""foiling the cracker": A survey of, and improvements to, password security," in *Proc. 2nd USENIX Security Workshop*, 1990.

[5] C. Kuo, S. Romanosky, and L. F. Cranor, "Human selection of mnemonic phrase-based passwords," in *SOUPS '06: Proc. 2nd Symposium On Usable Privacy and Security*, 2006.

[6] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *SOUPS '06: Proc. 2nd Symposium On Usable Privacy and Security*, 2006.

[7] J. Yan, A. Blackwell, R. Anderson, and A. Grant, "Password memorability and security: Empirical results," *IEEE Security and Privacy*, vol. 2, no. 5, 2004.

[8] J. Bunnell, J. Podd, R. Henderson, R. Napier, and J. Kennedy-Moffat, "Cognitive, associative and conventional passwords: Recall and guessing rates." *Computers & Security*, vol. 16, no. 7, 1997.

[9] J. A. Halderman, B. Waters, and E. W. Felten, "A convenient method for securely managing passwords," in *Proc. 14th International World Wide Web Conference*, 2005.

[10] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions," in *Proc. 14th USENIX Security Symposium*, 2005.

[11] S. Chiasson, P. C. van Oorschot, and R. Biddle, "A usability study and critique of two password managers," in *Proc. 15th USENIX Security Symposium*, 2006.

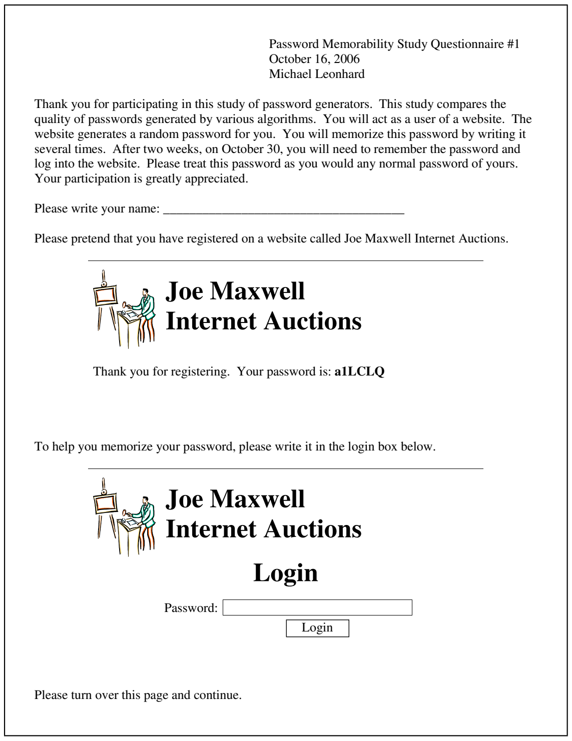
[12] M. D. Leonhard and V. N. Venkatakrishnan. (2007, March) A comparative study of three random password generators. [Online]. Available: http://tamale.net/pub/2007/pwdgen/

[13] (2007, March) Wordpress - blog tool and weblog platform. [Online]. Available: http://wordpress.org/

[14] A. G. Reinhold. (2006, October) Diceware passphrase home page. [Online]. Available: http://www.diceware.com/

[15] R. Ganesan and C. Davies, "A new attack on random pronounceable password generators," in *Proc. 17th NIST-NCSC National Computer Security Conference*, 1994.

APPENDIX



Password Memorability Study Questionnaire #1
October 16, 2006
Michael Leonhard

Thank you for participating in this study of password generators. This study compares the quality of passwords generated by various algorithms. You will act as a user of a website. The website generates a random password for you. You will memorize this password by writing it several times. After two weeks, on October 30, you will need to remember the password and log into the website. Please treat this password as you would any normal password of yours. Your participation is greatly appreciated.

Please write your name: ______________________________

Please pretend that you have registered on a website called Joe Maxwell Internet Auctions.

Joe Maxwell
Internet Auctions

Thank you for registering. Your password is: **a1LCLQ**

To help you memorize your password, please write it in the login box below.

Joe Maxwell
Internet Auctions

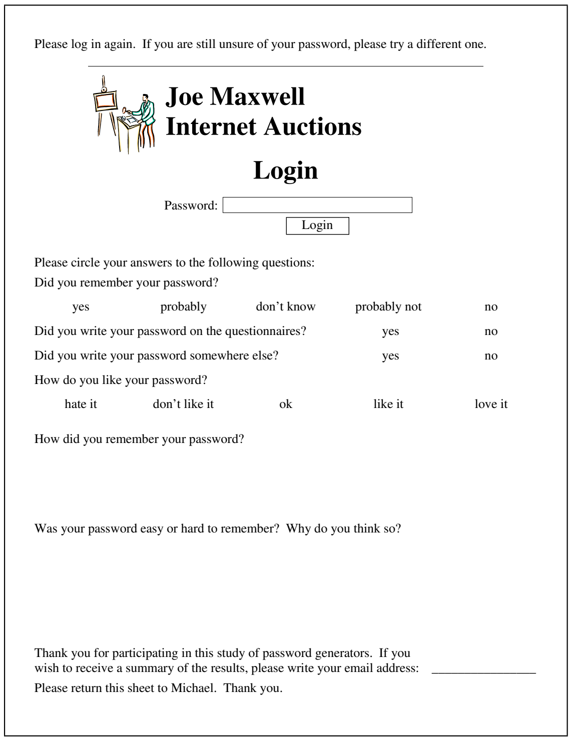
Login

Password: ____________

Login

Please turn over this page and continue.

Fig. 3. Front of First Questionnaire



Please log in again. If you are still unsure of your password, please try a different one.

Joe Maxwell
Internet Auctions

Login

Password: ____________

Login

Please circle your answers to the following questions:

Did you remember your password?

yes probably don't know probably not no

Did you write your password on the questionnaires? yes no

Did you write your password somewhere else? yes no

How do you like your password?

hate it don't like it ok like it love it

How did you remember your password?

Was your password easy or hard to remember? Why do you think so?

Thank you for participating in this study of password generators. If you wish to receive a summary of the results, please write your email address: ____________

Please return this sheet to Michael. Thank you.

Fig. 4. Back of Second Questionnaire