정 보 보 호 학 회 지
제28권 제1호, 2018. 2

Memorablity and Security of Different Passphrase Generation Methods

Simon S. Woo*, Jelena Mirkovic**

Abstract

Passphrases are considered to be more secure than passwords since they are longer than passwords. However, users choose predictable word patterns and common phrases to make passphrases memorable, which in turn significantly lowers security. While random passphrases appear to be stronger, surprisingly they are neither strong nor memorable. In this paper, we present the latest passphrase research, and introduce a new way to create a passphrase using mnemonics. Passphrase generation using mnemonics shows promising results in improving both strength and memorability.

Ⅰ. Introduction

Although textual passwords are widely used for today's user authentication, passwords are easily crackable if they are based on predicable patterns or if they are too short. One way to make passwords more secure is to make them longer. Longer passwords should be harder to guess by automated attacks, as the guessing space will be larger. A passphrase is one example of longer passwords, and is usually made by combining several words together. These words could be unrelated, e.g., "mother chicken apple", or form a sentence, e.g. "I love apple juice". Passphrases also tend to be more memorable than passwords, as they may contain expressions familiar to a user (e.g., verses of a favorite song) and follow grammatical rules [1-4]. However, there are several problems with passphrases. Underlying grammatical structure of passphrases and use of common phrases (e.g., verses from songs) lowers their security well below the security expected by length alone [5,6]. And if these patterns in passphrases are broken by forcing users to use system-generated passphrases, security increases but recall drops significantly[7]. Therefore, there is a trade-off between passphrase recall and security - it is hard to improve one without jeopardizing the other. In this paper, we examine and compare different ways to construct passphrases from prior research to better capture the trade-off. They are described below:

- **User chosen passphrase (UPass):** a user create a passphrase without other constraints (e.g: "I Love Apple Juice")
- **System-generated passphrase (SysPass):** a system chooses a random word from a dictionary and concatenates them to form a passphrase (e.g.: " Correct Horse Battery Staple")
- **Mnemonic-Guided Passphrase (MNPass)[20]:** a system generate a mnemonic alphabet and a user chooses a word start with each mnemonic (e.g: given mnemonics "ABALO", a user can create "Apples bread and lox order" as a passphrase). Each passphrase construction method has clear

본 연구는 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구입니다.(No. NRF-2017R1C1B5076474)
* 교신전자, 한국뉴욕주립대학교 컴퓨터과학과(SUNY, Korea) (simon.s.woo@sunykorea.ac.kr)
** USC Information Sciences Institute (mirkovic@isi.edu)

trade-offs in memorability and strength. In this paper, we compare each approach and provide recommendations for designing a better passphrase system for future research.

This paper is organized in the following way. In Section II, we present the relevant research on passphrases. In Section III, we define the passphrases, and provide the different passphase construction algorithms. In section IV, we describe the passphrase attack model. In Section V, we compare the security strength and memorability performance of each approach. Lastly, in Section VI, we offer some conclusion of passphrase construction.

Ⅱ. Related Work

Rao et al. [6] discovered that long passwords have a distinct grammatical structure. They analyzed part-of-speech (POS) tag sequences from the Brown Corpus [8] and found that the grammatical structure decreases search space for passwords by more than 50%. Veras et al. [9] explored semantic patterns of passwords and showed how these patterns can be used to greatly improve attack success. Bonneau and Shutova [10] studied short user-chosen passphrases (2 to 3 words), and showed that they are vulnerable to dictionary attacks, and that they have simple noun structure. Shay et al. [7] found that both system-generated passphrases and system-generated passwords are annoying to users and easy to forget. Cued-recall systems (e.g., [11,12,13]) have been proposed for graphical passwords, as summarized by Biddle et al. [14]. In these systems a user is shown an image or a set of images as a cue, and must recall which points on the image she clicked, or which images she selected, in order to authenticate. Bicakci and van Oorschot further propose grid-Words [15], textual, multi-word passwords that can be entered by selecting them from a dropbox or by locating them on a grid, which serves as a cue. Kuo et al. [16] researched the mnemonic passwords, which are derived as abbreviations of common phrases such as movie titles. Kuo et al. found 65% of mnemonic passwords via Google searches. User training has also been shown to improve password recall [9, 18], and it may use mnemonic techniques. Woo and Mirkovic[20] propose a novel way to construct passphrases using mnemonics. Their approach achieves a good balance between memorability and strength compared to users-chosen or system-generated passphrases. In this paper, we discuss the passphrase construction by Woo and Mirkovic[20] and report their findings, compared to other approaches.

Ⅲ. User-Chosen and System-Generated Passphrases

In this section, we first describe the user-chosen and system-generated passphrases and discuss some of their issues with respect to memorability and strength. Next, we present the new hybrid approach using mnemonics, which balances between user-chosen and system-generated passphrase approaches.

3.1. User-Chosen Passphrase (UPass)

Users can freely select a passphrase with any words or character combinations they wish, and many websites encourage users to create a passphrase to be longer than a typical password. However, there are several issues with user-chosen passphrases. As Bonneau and Shutova [10] showed. short user-chosen passphrases (2-3 words) are vulnerable to dictionary attacks, and it turns out that most users choose simple noun structures. For longer passphrases such as 5 or more words, the situation does not improve much. Woo and Mirkovic[20] show that users tend to choose combinations of popular words with underlying grammaritcal or list structures when creating longer passphrases. The examples are “My

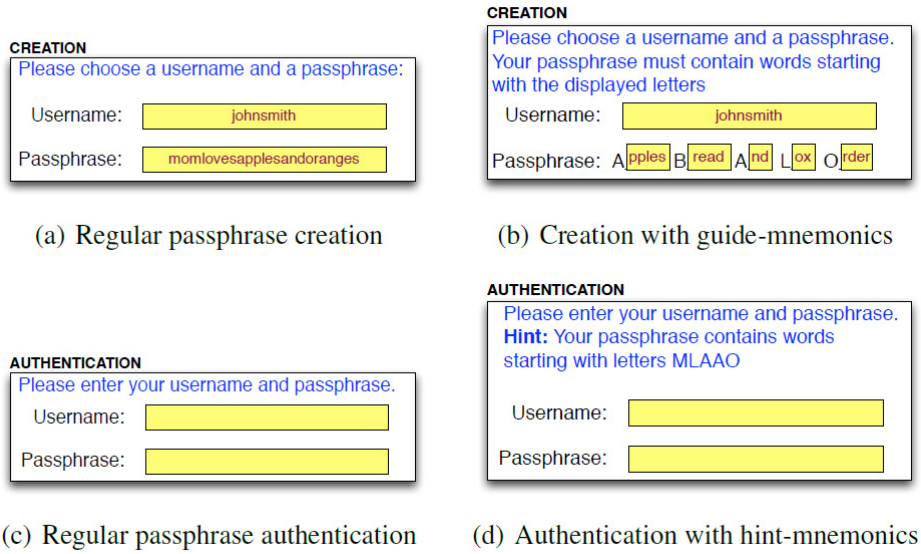
Cat Is Very Funny“ or ”Apple Banana Orange Grape Pear.“ Hence, these make strength of users-chosen passphases even weaker than a password, though they are longer than passwords. Another practical concern with a passphrase is that many websites enforce the maximum number of characters such as 15 to 20 characters. This only allows 3 to 5 words as a passphrase and ironically users cannot create a strong passphrase with this maximum letter constraint, as a system prohibits.

3.2. System-Generated Passphrase (SysPass)

The system-generated passphrase is composed of the randomly chosen words from a target dictionary. The most famous example of the system-generated passphrase is “Correct Horse Battery Staple” by XCDE[19]. The system-generated passphrases are not only difficult to remember since they completely do not has personal associations to users, but also they are not strong, if a size of dictionary is small. Shay et al. [7] and Woo and Mirkovic[20] confirmed that system-generated passphrases are not usable with extremely poor recall rate. Also, users feel that the system-generated passphrases were the least preferred among all three appraoches we discuss. The passphrase strength measured in entropy by Shay et. al[7] was between 29 to 39 bits with 4 word passphrases. This is much lower than a typical 53-bit password strength requirement. Hence, careful consideration must be made, when designing a system-generated passphrase.

3.3. Mnemonics-based Approach (MNPass)

Mnemonics can aid in improving memorability and strength a passphrase. In [20], they used mnemonics as first letters of passphrase words. They suggested the use of mnemonics in two ways. First, they can be used as user hints during authentication, to improve recall (aka recall cues [14]) - defining as **hint-mnemonics**. A user chooses a passphrase, and the system creates a hint-mnemonic and stores it with the passphrase. For example, a user may choose “Mom loves apples and oranges” and the resulting hint-mnemonic becomes “MLAAO”. At authentication, the system prompts the user for her passphrase, and displays the hint-mnemonic. Figure 1(c) illustrates regular authentication with no hints, and Figure 1(d) illustrates authentication with hint-mnemonics. However, use of mnemonics as hints will lower security. Passphrases, like passwords, are stored hashed and salted, but mnemonics must be stored in clear since they are displayed to users during authentication. Thus a statistical attacker can tailor his guessing to words starting with letters of the mnemonic, which greatly reduces the search space. The second possible use of mnemonics is during passphrase creation - **guide-mnemonics**. Because users tend to use common word sequences, popular phrases, and grammatical rules in passphrases [10, 16, 6], many passphrases can be guessed by mining these common patterns from public sources. Mnemonics can be used during creation to improve randomness of word choices in passphrases, and to reduce the reuse of passphrases across different accounts. Guide-mnemonics are generated by choosing letters from the alphabet according to some algorithm. A user is then prompted to generate a passphrase matching this mnemonic. Each passphrase word must start with one mnemonic letter, in order. For example, a system may generate a guide-mnemonic “ABALO” and the user may input a matching passphrase like “Apples bread and lox order”. Figure 1(a) illustrates regular passphrase creation, and Figure 1(b) illustrates creation with guide-mnemonics. They further allow extraneous passphrase words, which are not part of the mnemonic, because they may aid recall. For example, a user may input “Apples bread and the lox are ordered,” with “the” and “are” being extraneous words. If guide-mnemonic is also to be used as a hint-mnemonic, they adjust it before storing to reflect



(a) Regular passphrase creation (b) Creation with guide-mnemonics

(c) Regular passphrase authentication (d) Authentication with hint-mnemonics

〔Fig. 1〕 Different passphrase creation and auth. methods using hint- and guide-mnemonics

all the passphrase words (e.g., ABALO becomes ABATLAO). In addition, they propose MNPass(1) approach, which lets the system choose only one high entropy (uncommon) word for users to further improve strength.

Ⅳ. Attacker Models and Strength

In cracking passphrases, the most powerful attacker model is the **language model (LM) attacker**[20], which compiles probabilities for word sequences occurring together, and uses these to guide his guessing. The maximum probability of each passphrase can be calculated among all possible passphrases of the same word-length using an n-gram model. Then maximum probability can be converted into guess number using Monte Carlo Sampling as proposed in [21]. After obtaining a maximum probability of a passphrase from LM, they convert it into guess number by using Monte Carlo Sampling on a corpus of 100,000 randomly generated passphrases, as described in [21]. However, the problem with this approach is that attackers can also observe the hints. Therefore, more careful strength estimate is required, considering the distribution of mnemonics generation.

LM Adjustments for Mnemonics: When mnemonics are used as hints during authentication, this changes the language models as some words and some sequences become invalid. Hence, in order not to overestimate the strength, they adjust the language models for each passphrase by re-normalizing the word distributions. They define M be the hint-mnemonic for one given passphrase P_M, and adjust the corpus for P_M by keeping only those unigram, bigram, and trigram sequences, which contain the words starting with letters in M, and which follow the order of letters in M. Then they use these sequences to build heir language models and calculate the probability of P_M.

Ⅴ. Evaluation

5.1. User Study

The recall and strength of UPass, SysPass, and MNPass(1) are validated with the user study using Amazon Mechanical Turks. Their user study has the two parts: passphrase creation and authentication.

Passphrase creation, The short tutorial and

examples for each passphrase model were presented and then participants were asked to create one passphrase. All users were asked not to write down or copy their answers, and to rely on memory only.

Passphrase Authentication. Each user was asked to make two authentication visits, one after three days, and one after one week since passphrase creation. At most five trials were given to users to authenticate per passphrase and per visit. All users were asked not to paste their answers.

Participant and Passphrase Statistics. In total, there were 1,273 participants who created a passphrase. Out of 1,273 participants, 731 (57.47%) participants returned for the first authentication and 426 (58.3%) returned for the second authentication. Total of 393 participants completed both the first and the second authentication

5.2. Results

They considered recall successful if users matched the entire passphrase in its normalized form - with removed capitalization, punctuation and whitespaces. They denote this match criterion *exact* match. We also considered a *relaxed* match criterion, where we normalize nouns to their singular form, and verbs to their stem form using the Porter stemming algorithm, before both storing and authentication. They hypothesized, and our results prove, that this relaxed matching further improves recall, and does not greatly decrease strength against statistical attacks.

MNPass(1) Recall Comparable to UPass Recall. Mnemonic-guided passphrase creation may jeopardize personal significance of passphrases to the user, and thus impair recall. However, the result shows that MNPass(1) is a little lower than that of UPass (69.3% vs 71.4% after three days, 67.7% vs 69.6% after seven days) as shown in Table 1. On the other hand, the generation of all words by the system drastically lowers recall yielding only around 19% to 28%.

〔Table 1〕 Recall Performance (%)

	exact	relax	exact	relax
Recall interval	3 day	7 day	3 day	7 day
UPass	71.4	69.6	76.8	73.2
SysPass	26.8	18.9	28.7	19.6
MNPass(1)	69.3	67.7	75.8	69.3

〔Table 2〕 Guess Number Strength with LM Attacker

	exact	relax
UPass	1.2×10^{10}	8.8×10^{9}
SysPass	3.9×10^{17}	3.9×10^{17}
MNPass(1)	1.3×10^{16}	5.4×10^{15}

MNPass(1) Improves Strength. The strength of MNPass(1) is much stronger than UPass and close to SysPass. If one difficult word is generated from a system in MNPass(1), it significantly improves the strength performance.

Relaxed Matching Is Acceptable. Relaxed matching lowers security, because more of the attacker's guesses lead to successful authentication. However, relaxed matching lowers the strength by at most 10 , and thus has an acceptable security cost, while greatly improving recall

Ⅵ. Conclusion

It is challenging to create a passphrase approach, which has both a high recall and a high strength. The use of mnemonics as authentication hints significantly improves recall, because it helps users remember which words they chose during passphrase creation. Mnemonics can further be used to guide passphrase creation, which reduces use of common phrases and improves strength. By allowing the system to choose one word in a passphrase, strength can be further improved. We thus believe mnemonics are a promising technique to improve user authentication.

References

[1] KEITH, M., SHAO, B., AND STEINBART, P. A Behavioral Analysis of Passphrase Design and Effectiveness. Journal of the Association for Information Systems 10, 2 (2009),

[2] KEITH, M., SHAO, B., AND STEINBART, P. J. The Usability of Passphrases for Authentication: An Empirical Field Study. International journal of human-computer studies 65, 1 (2007), 17-28.

[3] SPECTOR, Y., AND GINZBERG, J. Pass-sentence - a New Approach to Computer Code. Computers & Security 13, 2 (1994), 145-160.

[4] ZVIRAN, M., AND HAGA, W. J. A Comparison of Password Techniques for Multilevel Authentication Mechanisms. The Computer Journal 36, 3 (1993), 227-237.

[5] KUO, C., ROMANOSKY, S., AND CRANOR, L. F. Human Selection of Mnemonic Phrasebased Passwords. In Proceedings of the 2006 Symposium on Usable Privacy and Security, pp. 67-78.

[6] RAO, A., JHA, B., AND KINI, G. Effect of Grammar on Security of Long Passwords. In Proceedings of the third ACM conference on Data and application security and privacy(2013), pp. 317-324.

[7] SHAY, R., KELLEY, P. G., KOMANDURI, S., MAZUREK, M. L., UR, B., VIDAS, T., BAUER, L., CHRISTIN, N., AND CRANOR, L. F. Correct Horse Battery Staple: Exploring the Usability of System-assigned Passphrases. In Proceedings of the 2012 Symposium on Usable Privacy and Security, p. 7.

[8] FRANCIS, W. N., AND KUCERA, H. Brown corpus manual. Brown University (1979).

[9] VERAS, R., COLLINS, C., AND THORPE, J. On semantic patterns of passwords and their security impact. In NDSS (2014).

[10] BONNEAU, J., AND SHUTOVA, E. Linguistic Properties of Multi-word Passphrases. In Financial Cryptography and Data Security. Springer, 2012, pp. 1-12.

[11] CHIASSON, S., FORGET, A., BIDDLE, R., AND VAN OORSCHOT, P. C. Influencing Users Towards Better Passwords: Persuasive Cued Click-points. In Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1 (2008), British Computer Society, pp. 121-130.

[12] CHIASSON, S., VAN OORSCHOT, P. C., AND BIDDLE, R. Graphical Password Authentication Using Cued Click Points. In European Symposium on Research in Computer Security(2007), Springer, pp. 359-374.

[13] DAVIS, D., MONROSE, F., AND REITER, M. K. On User Choice in Graphical Password Schemes. In USENIX Security Symposium (2004), vol. 13, pp. 11-11.

[14] BIDDLE, R., CHIASSON, S., AND VAN OORSCHOT, P. C. Graphical passwords: Learning from the first twelve years. ACM Computing Surveys (CSUR) 44, 4 (2012), 19.

[15] Bicakci, Kemal, and Paul C. van Oorschot. "A multi-word password proposal (gridWord) and exploring questions about science in security research and usable security evaluation." Proceedings of the 2011 New Security Paradigms Workshop. ACM, 2011.

[16] KUO, C., ROMANOSKY, S., AND CRANOR, L. F. Human Selection of Mnemonic Phrasebased Passwords. In Proceedings of the 2006 Symposium on Usable Privacy and Security, pp. 67-78.

[17] BLOCKI, J., KOMANDURI, S., CRANOR, L., AND DATTA, A. Spaced Repetition and Mnemonics Enable Recall of Multiple Strong Passwords. arXiv preprint arXiv:1410.1490 (2014)

[18] DAS, S., HONG, J., AND SCHECHTER, S. Testing Computer-Aided Mnemonics and

Feedbackfor Fast Memorization of High-Value Secrets. Proceedings of the 2016 Usable Security Workshop.
[19] Password Strength xkcd, https://xkcd.com/936/
[20] Simon S. Woo and Jelena Mirkovic. “Improving Recall and Security of Passphrases Through Use of Mnemonics”, Proceedings of the 10th International Conference on Passwords (Passwords), Bochum, Germany, 2016.
[21] DELL’AMICO, M., AND FILIPPONE, M. Monte Carlo Strength Evaluation: Fast and Reliable Password Checking. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (2015), ACM, pp. 158-169.

〈저 자 소 개〉

우 사이먼 (Woo. Simon)
정회원
2003년 : BSEE, University of Washington, Seattle, USA
2005년 : MSECE, University of California, San Diego, La Jolla, USA
2017년 : Ph.D. in CS, University of Southern California, Los angeles, USA
2017년～현재 : 한국뉴욕주립대학교 (SUNY, Korea) 컴퓨터과학과 조교수
관심분야 : 통신공학, 정보보호

Mirkovic. Jelena
1998년 : BSEE, University of Belgrade, Serbia
2000년 : MSCS, UCLA, USA
2003년 : Ph.D. in CS, UCLA, USA
2010년～Present : Research Faculty at USC/ISI
관심분야 : Computer, Network, and Usable Security