IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 7, 2019, accepted August 15, 2019, date of publication August 23, 2019, date of current version September 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937030

Alphapwd: A Password Generation Strategy Based on Mnemonic Shape

JIANHUA SONG[1,2,3,4], DEGANG WANG[1], ZHONGYUE YUN[1], AND XIAO HAN[1]

[1]School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China
[2]Engineering and Technical Research Center of Hubei Province in Educational Informatization, Wuhan 430062, China
[3]Engineering and Technical Research Center of Hubei Province in Software Engineering, Wuhan 430062, China
[4]Engineering Research Center of Hubei Province in Intelligent Government Affairs and Application of Artificial Intelligence, Wuhan 430062, China

Corresponding author: Degang Wang (wangpei_hubu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 51472079 and Grant 61702377, in part by the Technology Innovation Special Program of Hubei Province under Grant 2018ACA13, and in part by the Hubei Province Project of Key Research Institute of Humanities and Social Sciences at Universities (Research Center of Information Management for Performance Evaluation).

ABSTRACT Password-based authentication is the first line of defense of most information systems. Password security concerns the security of the whole information system. Therefore, administrators will formulate corresponding password strategies to help users improve the security and usability of passwords. Several common password composition strategies are analyzed. Aiming at the problem that traditional password strategies cannot take security and usability into account, a new password generation strategy based on mnemonic shape, Alphapwd, is proposed. Alphapwd combines the order of writing strokes of letters with password generation to help users create safe and memorable passwords. The designed experiment compares the security of Alphapwd-based passwords with three leaked password sets. The result shows that Alphapwd-based password is generally stronger than real password sets in resisting unknown attacks. In addition, by analyzing the passwords generated by Alphapwd and KbCg (Keyboard Change), SpIns (Special Character Insertion), it can be found that the security of Alphapwd password is better than that of KbCg mnemonic password, and Alphapwd password is easier to remember than SpIns mnemonic password.

INDEX TERMS Password generation strategy, password strength, mnemonic passwords.

I. INTRODUCTION

There are many authentication methods, such as password-based and graphics authentication (based on what you know), hardware authentication(based on what you have), biometric-based authentication(based on what you are) [1]–[3]. However, password-based authentication will still be the mainstream authentication method for a long time in the future [4], because password-based authentication is simple and easy to use, low cost, easy to manage, while other authentication methods have many problems, such as high cost, difficult to deployment, privacy disclosure, and so on. Password-based authentication methods are applied to various information systems, such as account login, data encryption and so on. Once the password is cracked or leaked, it will cause users' personal information leakage, economic property losses, confidential data theft and other serious consequences. With the rapid development of Internet technology, each user needs to manage more and more passwords, and in general, strong passwords are difficult to remember. Therefore, users may choose name, date of birth, telephone number, hobbies, or a combination of them as passwords, which provides the possibility for targeted password attacks and password reuse attacks based on personal information [5], [6]. Password security and usability have become a contradiction, users are caught between them and suffer from both of them. To solve these problems, a password generation strategy based on mnemonic is proposed, which is safe and easy to remember.

The remainder of this paper is organized as follows. Section II reviews the related work of password composition strategy. Section III introduces several commonly used mnemonic password composition strategies. Section IV proposes a password generation strategy based on mnemonic shape with good usability, and experiments are designed to evaluate its security and usability. Section V summarizes the work done in this paper.

II. RELATED WORK

Password composition policies are used to help users to create passwords. Komanduri *et al.* [7] studied the password

The associate editor coordinating the review of this article and approving it for publication was Mamoun Alazab.

strength, user behavior and user psychology of four password composition strategies in different scenarios. The experimental results show that the password entropy created by Condition basic16 policy (The required password length should not be less than 16 characters) is the largest. Condition dictionary8 policy (Required password length should not be less than 8 characters and dictionary words should not be included) can effectively prevent users from creating passwords that can be easily cracked by heuristic methods, but dictionary checking will make users feel frustrated. Shay *et al.* [8] studied the security and usability of composition strategies with long-length by online experiments. The results show that 3class12 policy (Required password length should not be less than 12 characters and each password should contain at least three character types) has better usability and 2word16 policy (Required password length should not be less than 12 characters and each password should contain at least two words) has stronger security. Yajun *et al.* [9] used Yahoo, Tianya and Linkedin leaked password sets to analyze the security of real password sets in three cases: no password composition strategy, basic6 strategy (password length is not less than 6 characters) and 2class6 policy (Password length is not less than 8 characters and each password contains at least two different kinds of characters in upper and lower case letters, numbers and symbols). It is found that none of the above three password composition strategies can help users to create strong passwords.

Yang *et al.* [10] studied the usability and security of six variants of mnemonic password generation strategy. The results show that MneGenEx is easy to lead to weak password, MnePerEx strategy is easy to create strong password, and by giving sentences, they can crack more than half of the passwords created by users in 5 to 10 guesses. Kiesel *et al.* [11] established a very large mnemonic corpus and used it to evaluate the security of sentence-based mnemonic passwords. Experimental results show that mnemonic passwords created with only lowercase letters are as strong as seven-length ones created with printable ASCII characters. The experimental effect of the mnemonic passwords with less complexity in the offline attack scenario is lower than expected, and the longer mnemonic password performs better in the offline attack scenario, but this is not necessarily the case in the online attack scenario. Compared with the passwords generated by dictionary sampling, mnemonic passwords can achieve the same password distribution intensity with fewer characters in offline attack scenarios. Bei *et al.* [12] analyzed the strength of four mnemonic passwords, SenSub, KbCg, UsForm and SpIns. The study shows that in the unknown attack scenario, the strength of the four mnemonic passwords is higher than that of the two control groups (from the password sets leaked by websites 178 and phpBB). The password distribution security of UsFom is the highest, and the password created by SenSub is stronger than the other three mnemonic passwords in the known attack scenario. Based on the fact that graphics are easier to remember than strings, Guo *et al.* [13] proposed a user-friendly password composition strategy, Optiwords, and compared the security and usability of Optiwords with other popular password policies. The results show that compared with basic8 or 3class8, Optiword has no significant difference in memory ability, and the password strength of Optiwords is higher than that of basic8 and 3class8. In terms of usability, Optiwords is better than Random8.

The PTP password generation strategy proposed by Forget *et al.* [14] suggests that random characters can be inserted into the password set by the user to improve the security of the password. It is found that the PTP password generation strategy can only improve the security of the password in a small increase, because in order to create the password which is easy to remember, the password entered in advance is generally very weak. Huh *et al.* [15] proposed a password generation strategy for generating a secure random password from the system, and the user changed some of the letters in the password. The results show that with the increase of the number of characters replaced by the user, the memorability of the password increases slightly. Compared with the password under the general password strategy, the cracked rate of this scheme is reduced by 21%, but it's still hard for users to remember.

III. MNEMONIC PASSWORD STRATEGY

Since common password composition strategy simply limits the character set diversity of passwords, some scholars have advanced the mnemonic password strategy on the basis of the general password composition strategy. The purpose of the mnemonic password policy is to help users create passwords that are secure and easy to remember.

The similarities between MneGenEx, MnePerEx, MnePer, MneEx, MneSchEx, and MneYanE are that the user selects sentences or phrases that contain at least eight words (It is safer to choose what makes sense to ourselves and is unlikely to be used by others), and then replaces each word with numbers, letters, and special symbols, usually using the first letter of each word. For example, using the MneGenEx strategy, select the sentence "Four score and seven years ago our fathers brought forth on this continent, then "Four" =>'4' ('4' replace "Four"), "and"=>'&', "seven"=>'7', "forth"=>'4', the rest of the words are replaced by the first letter of the word.Then we get the password "4s&7yaofb4otc".

The KbCg password policy allows user to select an easy-to-remember password as pawd1 firstly, then the user moves one or more keys on the keyboard as a password based on pwd1 to the top left, top right, bottom left, and bottom right.For example, use the word "handsome", and choose to move a key to the top left, the password will be "yqhew9j3".

The UsForm(Using a Formula) password policy allows user to select an easy-to-remember mathematical formula (such as the addition formula), and then select several numbers to calculate the password. For example, if we select the addition formula with numbers 1, 2, 5, then the password can be "1+ two+five=8" or "1 +two+five=eight".

SpIns lets the user select a password that is easier to remember, then choose an easy-to-remember insertion method, inserting special characters to the password to get the final password. For example, if we use the word "handsome" and insert some symbols to the word, then the password will be like "h@and!Some!".

The Optiwords strategy allows user to select the pattern he or she likes, and then simulates the pattern in the printable character area of the keyboard, where the user selects the characters on the key through which the pattern passes as the password. An example of creating a password using Optiwords password composition strategy is shown in Figure 1, the dot in the figure indicates that the key was typed with the "Shift" key, and we can get the password "%rDfGthbnj" by selecting all characters passing through the pattern in left-to-right, top-to-bottom order.

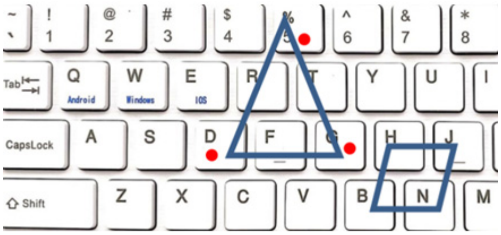


FIGURE 1. Optiwords example.

Mnemonic characters do help users create passwords that are easy to remember, but what about security? Das *et al.* [16] studied the user password reuse from the leaked password sets through a series of text similarity algorithms. It was found that users only made simple changes from one site to another site, such as sequence change ("qwq233"->"233qwq"), capitalize the word("hello233"->"Hello233"), Leet change ("password"->"p@$$word"), password reversal("password->"drowssap"). Then Anupam Das et al. developed a cross-station password guessing algorithm, the success rate is 16% higher than the standard password guessing algorithm when each password is guessed up to 100 times. Schweizer *et al.* [17] proposed that using keyboard mode can create passwords that appear to be random and very easy to remember, such as "1qazv!QAZ2wsx@". By using visualization technical to collect data and assist in pattern classification, Dino Schweizer et al. have successfully identified 18.2% keyboard mode passwords in a true password set that is not covered by traditional dictionaries. Chou *et al.* [18] proposed a AP framework to describe the adjacent mode and parallel mode in keyboard mode, and then generated a password set to crack the password. The experimental results show that the size of password set generated by using AP framework is $2^{44.47}$ times smaller than that of brute force cracking, and the guessing set with AP mode password is 114% more successful than that guessing set without AP mode password. On the basis of probabilistic context-free grammar (PCFG) [19] attack algorithm, Houshmand *et al.* [20] further add keyboard mode to password structure to carry out attack experiment. The experimental results show that their algorithm is about 20% more efficient than the original PCFG-based model attack.

IV. ALPHAPWD PASSWORD GENERATION STRATEGY BASED ON MNEMONIC SHAPE

A. THE BASIC IDEA OF ALPHAPWD PASSWORD GENERATION STRATEGY

Through the above analysis, it's known that the password generated based on mnemonic policy is easy to remember, but the security is not very ideal. Random password can solve the security problem, but the usability is greatly reduced. In the challenging of more and more advanced password attack algorithms, how to generate passwords that are secure enough and easy to remember?

The graphic-based keyboard mode password strategy proposed by Guo *et al.* [13]——Optiwords has opened up a new direction for the mnemonic password strategy, but there is still a certain problem that the usability and security cannot be balanced: In the Optiwords password strategy, users need to remember the patterns they choose. The use of complex patterns is highly secure, but not easy to remember. The use of simple patterns will reduce the security. So is there a better password strategy to solve this problem? It is well known that both letters and Chinese Pinyin have a certain stroke order when writing. Take the simple alphabet writing as an example, the stroke order of the letters 'A'/'a' in Figure 2, which can be used to generate passwords. In Figure 3, it simulates the writing process of characters 'A' and 'a' on the keyboard. In the process of writing 'A', the user successively go through keys '4', 'e', 's', '4', 'r', 'f', 'e', 'r' on the keyboard. Then we can get the string "4es4rfer" ("4es" is the first stroke of letter A, "4rf" is the second stroke, "er" is the third stroke) by taking the characters on the keys respectively. In the same way, after simulating the writing order of 'a', then we can get the string "87yhj8ik" ("87yhj" is the first stroke. "8ik" is the second stroke). Based on this, we propose a password generation strategy based on mnemonic shape ——Alphapwd. The basic idea of Alphapwd is as follows:

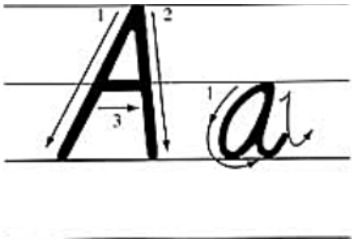


FIGURE 2. Stroke order of 'A'/'a'.

1) Alphapwd password generation strategy requires that the generated password contains at least two types of uppercase letters, lowercase letters, Numbers and symbols.

2) By simulating the writing order of the letters on the keyboard, the symbols represented by the keys are taken as the password.

TABLE 1. A letter substitution scheme conforming to alphapwd strategy.

Mnemonic	Sequence	Mnemonic	Sequence	Mnemonic	Sequence
a	87yhj8ik	s	98ikj	K	YHNhuhm
b	4eserds	t	uio8ikl	L	YGVb
c	54er	u	7ui88io	M	CFTgbhujm
d	iuhj9ij	v	8iko0	N	YGVyhnji
e	ui87yhj	w	4r5t6	O	(*ui9
f	87ygftyu	x	8i9u	P	YHNyujh
g	poklpl,m	y	t6y7yg	Q	76tghu7j
h	9ijiok	z	78uhj	R	6yh67uyj
i	6yh	A	4es4rfer	S	987ujmnb
j	7ujh	B	5tg56yttyhg	T	%^&6yh
k	4rfr5rg	C	76TGH	U	5tghju7
l	7uji	D	UHBuikmnb	V	YHNji
m	i9o0p	E	yhnmyuhj	W	TGBhujmko
n	ujjuik	F	YGVyugh	X	&UJ8uh
o	09io0	G	87yhjik	Y	7u9ijn
p	okmoplk	H	UHBijnhj	Z	WERrdxXCV
q	76yu7uj	I	7898uhghj		
r	y7ujju78	J	8ikj		

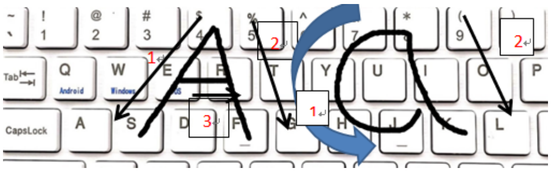


FIGURE 3. Simulation of 'A'/'a' on keyboard.

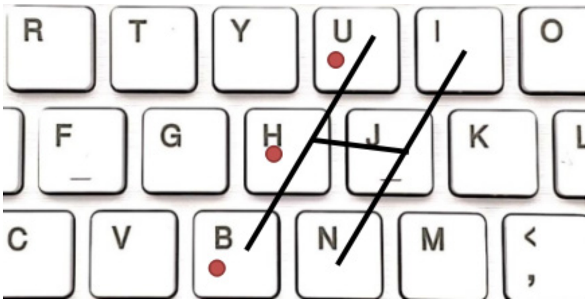


FIGURE 4. Simulation of 'H' on keyboard.

3) The user can choose whether to use all the key symbols as the password or the selection part (for example, the symbol on the key is selected at intervals without affecting the outline of the letter).

4) The users only need to remember the mnemonic they chose and its starting position on the keyboard, then use it to enter the password.

B. A CASE STUDY OF ALPHAPWD

We give a character substitution scheme of 52 common letters with the Alphapwd strategy in Table 1 (Of course, there are many other letters replacement schemes). How to get the substitution scheme of character 'A' and 'a' has been discussed in section A. In Figure 4, the character "H" can be replaced with "UHBijnhj", where "UHB" is the first stroke of H, "ijn" and "hj" are the second and third strokes respectively (the dot in the figure indicates that the key was typed with the "Shift" key). Other characters are replaced in a similar way.

A password generation example presented here, with the name pinyin initials "wdg", take in the written order of Figure 5 we can get the generated password "4r5t6iuhj9ijpoklpl,m" (w=>"4r5t6", d="iuhj9ij", g=> "poklpl,m"), the password contains letters, numbers, symbols, looks like a random string. If brute force algorithm is used to crack the password, the search space will be 95^{20}. However, just three mnemonic characters and its starting position need to be remembered ('w' starting position is '4', 'd' starting at 'i' and 'g' starting at 'p'), then the password can be entered quickly and accurately.

Alphapwd password strategy has more flexibility than the common keyboard mode password strategy. New passwords can be obtained by translating the position of the

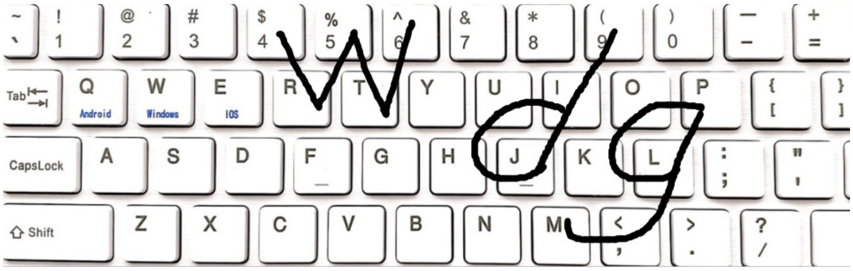

~ ` ! 1 @ 2 # 3 $ 4 % 5 ^ 6 & 7 * 8 (9) 0 — – + =
Tab Q Android W Windows E IOS R T Y U I O P { [}]
CapsLock A S D F G H J K L : ; " '
Shift Z X C V B N M < , > . ? /

FIGURE 5. Simulation of 'w','d','g' on keyboard.

written letter on the keyboard, or expanding or narrowing the writing range. For example, "trdf6tf" shifts a key to the right to become "uygh8uh", and when a user is asked to periodically update a password, the user can choose this transformation to change the password without changing the mnemonic of his choice. The user can also just replace letters partially, such as the string "wang", only replace the character 'a' ('a'->"ewazxdeedc"), and the password is "wewazxdeedcng".

C. ALPHAPWD-BASED PASSWORD STRENGTH EXPERIMENT

Common password strength evaluation methods include rule-based evaluation, pattern matching evaluation and attack algorithm evaluation [4]. PCFG and Markov attack algorithm are adopted in this section [19], [21] to evaluate Alphapwd-based password's security. A password is considered weak if it requires less than 10^6 guesses when successfully be cracked, and strong if it requires more than 10^{14} guesses [22]. Dell 'Amico and Filippone [23] proposed a novel and effective method for calculating the number of password guesses – Monte Carlo method, which uses probability method to quickly calculate the approximate number of guesses needed for a password to be cracked.

1) TRAINING PROCESS

In order to evaluate the security of Alphapwd password strategy, two groups of experiments are designed. In experiment 1, 1000 passwords are generated with Alphapwd strategy, and 1000 passwords are selected randomly from the leaked password sets of websites 178, 12306 and 7k7k respectively as three control groups. These four groups of password sets are used as test data. In addition, in order to be fair, password sets from leaked 178,12306 and 7k7k are not used as training sets, and about 142.06 million, 6.42 million and 4.67 million passwords are selected as training sets from CSDN, Renren and Rockyou leaked password sets respectively. PCFG algorithm and second-order Markov model are used in the training process. Firstly, the password dictionary training sets are generated, and the password to be measured is given. Secondly, the probability of the password was calculated with the generated password dictionary. Finally, the Monte Carlo method is used to calculate the number of guesses needed when it is cracked through the probability of the password. In experiment 2, we used the two mnemonic password strategies mentioned in section III, KbCg and SpIns mnemonic strategy, to generate 100 passwords respectively, and then randomly select 100 passwords from the 1000 passwords generated by Alphapwd in experiment 1. These three sets of passwords are used as test data, about 6.42 million passwords of leaked from CSDN are used as training sets. The security of the password strategies for Alhapwd, KbCg and SpIns was evaluated in the same way as in experiment 1.

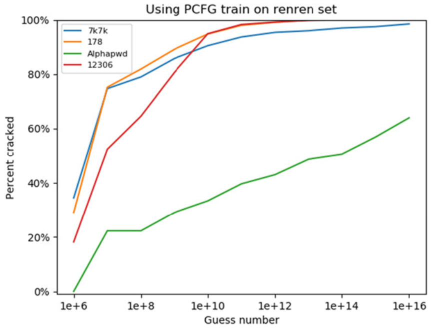


FIGURE 6. Using PCFG algorithm to train Renren password set to crack passwords.

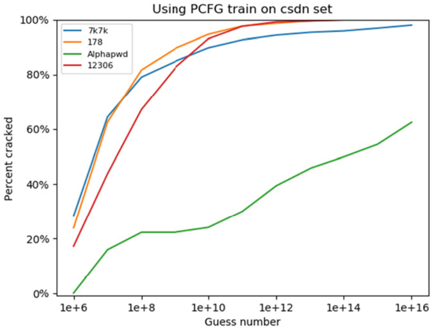


FIGURE 7. Using PCFG algorithm to train CSDN password set to crack passwords.

2) EXPERIMENTAL RESULTS AND ANALYSIS

Figure 6, 7 and 8 show the cracking of four groups of test set passwords when using PCFG algorithm to train Renren, CSDN and Rockyou training sets, while

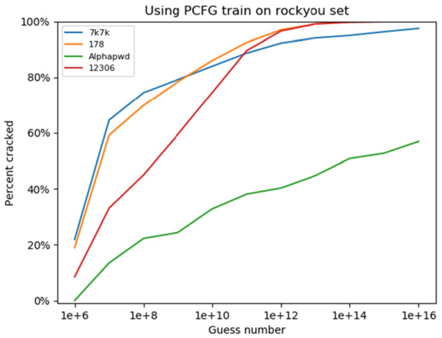


FIGURE 8. Using PCFG algorithm to train Rockyou password set to crack passwords.

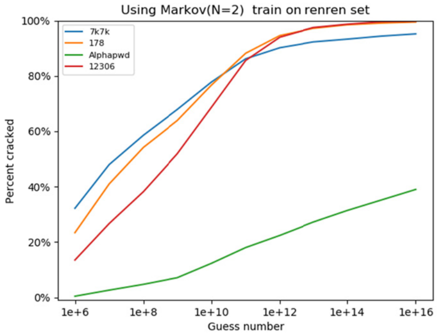


FIGURE 9. Using 2-Markov model to train Renren password set to crack passwords.

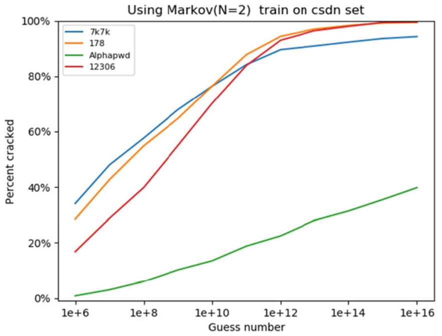


FIGURE 10. Using 2-Markov model to train CSDN password set to crack passwords.

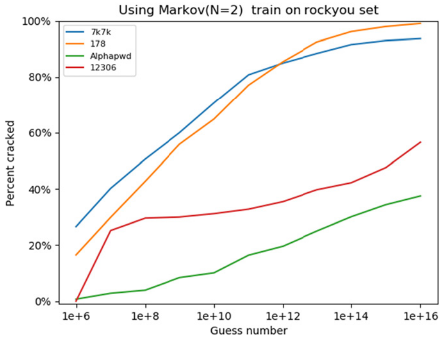


FIGURE 11. Using 2-Markov model to train Rockyou password set to crack passwords.

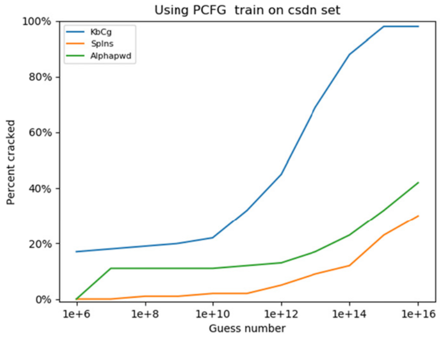


FIGURE 12. Using PCFG algorithm to train Renren password set to crack three kinds of mnemonic passwords.

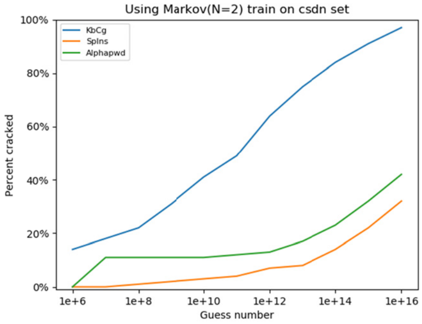


FIGURE 13. Using 2-Markov model to train Renren password set to crack three kinds of mnemonic passwords.

Figure 9, 10 and 11 show the cracking of four groups of test set passwords using second-order Markov model respectively. Figure 12 and 13 show the cracking of three mnemonic passwords using the PCFG algorithm and the second-order Markov model respectively. The vertical coordinates of Figure 6 to Figure 13 represent the percentage of password cracking in the test set, and the horizontal coordinates represent the number of guesses.

The results showed that by using PCFG to crack passwords, 23.3% of the Alphapwd test set were cracked under 10^8 guessing times, while 74.5-79.1%, 69.9-81.9% and 45%-67.2% of the 7k7k, 178 and 12306 control groups were cracked respectively. Under 10^{14} guessing times, 49.9%-50.8% passwords of Alphapwd test set were cracked, while 95.0%-97.0%, 99.9%-100% and 99.7%-100% passwords of 7k7k, 178 and 12306 control groups were cracked respectively. Using the second-order Markov model to crack passwords, only 3.9-6% passwords in the Alphapwd test set were cracked under the 10^8 guessing times, while 50.6%-58.6%, 42.7%-54.9% and 29.6%-39.9% passwords in the 7k7k, 178 and 12306 control groups were cracked respectively. Under 10^{14} guessing times, the passwords of Alphapwd test set were cracked by 30.1% to 31.4%, and the passwords of websites 7k7k, 178 and 12306 control groups were cracked by 91.6% to 93.3%, 96.2% to 98.5% and 42.2% to 98.7% respectively. When using the second-order Markov model to crack passwords, Rockyou password set as the training set to crack 12306 1000 passwords, did not work very well. Under 10^{14} guesses, the success rate of cracking KbCg, Alphapwd and SpIns passwords, using PCFG algorithm, was 98%, 42% and 30% respectively, while in Markov model, the success rate was 97%, 42% and 32% respectively.

By comparing Figure 6 and 9, Figure 7 and 10, Figure 8 and 11 respectively, it can be found that the effect of using PCFG algorithm is better than that of second-order Markov model. Combining the results of PCFG algorithm and second-order Markov model, it is found that the ability of Alphapwd password set to resist unknown attacks is the best, and the ability of 12306 password set to resist unknown attacks is slightly better than that of 7k7k and 178. Perhaps 12306 is a website that people often use and contains privacy information such as ID number, so the password set from 12306 is stronger. In the further analysis, it is found that the length of the password guessed by the Alphapwd part is usually short under the number of 10^{14} guesses, so when creating the password with Alphapwd, it is suggested that the password length should not less than 8 and contains at least two kinds of uppercase letters, lowercase letters, numbers and special symbols, and other password generation strategies can also be mixed to further increase the security of the password. Comparing the security of SpIns, Alphapwd and KbCg, it is found that the three mnemonic password strategies generated by SpIns, Alphapwd and KbCg in sequence from stronger to weaker. Although the password strength of SpIns is slightly stronger than that of Alphapwd, it is difficult to choose an easy-to-remember method of inserting special characters, however, Alphapwd does not have this problem.

D. ALPHAPWD-BASED PASSWORD USABLITY EXPERIMENT

We recruited 22 students to participate in our experiment for the usability testing. The experiment is divided into two stages. In the first stage, participants are required to create password using the Alphapwd strategy, and then asked to recall their passwords. In the second stage, participants are asked to return to the lab and recall their passwords after two days. Record the time taken to create the password and the time to recall the password. Also record the success rate of password recall.

In the first stage, it took about 10 seconds for each participant on average to create a password, and only two participants failed to recall the password correctly. Each participant recalled his or her password in 6.5 seconds on average. In the second stage, 81.81% of participants successfully recalled their passwords. Each participant recalled the password in 11 seconds on average. Compared with the success rate of short-term recall, the success rate of long-term recall did not decrease significantly.

V. CONCLUSION

In this paper, several common password composition strategies are introduced, and their security and usability are analyzed. In order to solve the problem that the password freely chosen by user is often weak, and the password generated by the system is secure but difficult to remember, Alphapwd, a password generation scheme based on mnemonic shape, is proposed. And it is verified that the password generated by Alphapwd has better ability to resist unknown attacks than the real passwords commonly used. Alphapwd password generation strategy also has the following advantages:

1) The generated password is easy to remember. It can be entered accurately and quickly by simply contrasting the layout of the keyboard (suitable for both computer keyboard and mobile keyboard).

2) It is easy to partially replace the letters to extend password length. When the password length entered by the user fails to meet the requirement of the minimum password length of password policy, Alphapwd strategy can be used to expand the password length. For example, when the password length entered by the user is "iloveyou520" and the website requires the password length to be at least 16, then use "rfvgy" instead of the character 'v', the final password is "ilorfvgyeyou520".

3) The user can update the password without changing the password mnemonic. For example, the user chooses the string "wdg" as the mnemonic to generate a password "1q2w3trdf6tfiuhjijnb" ('w'=>"1q2w3", 'd'=>"trdf6tf", 'g'=>"iuhjijnb"), we can converts "trdf6tf" to "uygh8uh" when updating the password, and the password becomes "1q2w3uyghiuhjijnb".

Having a password with high security does not mean that the system can be well protected, users also need to have a strong sense of security and develop good password usage habits. It is believed that more secure and memory-friendly password composition policies or new authentication methods will be proposed in the future.

REFERENCES

[1] Z. Zhao, G.-J. Ahn, and H. Hu, "Picture gesture authentication: Empirical analysis, automated attacks, and scheme evaluation," *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 4, p. 14, 2015.

[2] A. A. Darwish, W. M. Zaki, O. M. Saad, N. M. Nassar, and G. Schaefer, "Human authentication using face and fingerprint biometrics," in *Proc. 2nd Int. Conf. Comput. Intell., Commun. Syst. Netw.*, Jul. 2010, pp. 274–278.

[3] C. Wang, "The solution design using USB key for network security authentication," in *Proc. 4th Int. Conf. Comput. Intell. Commun. Netw.*, Nov. 2012, pp. 766–769.

[4] P. Wang, D. Wang, and X. Huang, "Advances in password security," (in Chinese), *J. Comput. Res. Develop.*, vol. 53, no. 10, pp. 2173–2188, 2016.

[5] Y. Li, H. Wang, and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in *Proc. IEEE 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[6] D. Wang, Z. Zhang, P. Wang, J. Yan, and X. Huang, "Targeted online password guessing: An underestimated threat," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1242–1254.

[7] 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 *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 2595–2604.

[8] R. Shay, S. Komanduri, A. L. Durity, P. Huh, M. L. Mazurek, S. M. Segreti, B. Ur, L. Bauer, N. Christin, and L. F. Cranor, "Can long passwords be secure and usable?" in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2014, pp. 2927–2936.

[9] Y. Guo, B. Ye, and W. Zhou, "Security analysis of user's real password under different password composition strategies," (in Chinese), *Netinf. Secur.*, vol. 19, no. 6, pp. 37–44, 2019.

[10] W. Yang, N. Li, O. Chowdhury, A. Xiong, and R. W. Proctor, "An empirical study of mnemonic sentence-based password generation strategies," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1216–1229.

[11] J. Kiesel, B. Stein, and S. Lucks, "A large-scale analysis of the mnemonic password advice," in *Proc. NDSS*, 2017, pp. 1–13.

[12] Y. Bei, G. Yajun, Z. Lei, and G. Xiaowei, "An empirical study of mnemonic password creation tips," *Comput. Secur.*, vol. 85, pp. 41–50, Aug. 2019.

[13] Y. Guo, Z. Zhang, and Y. Guo, "Optiwords: A new password policy for creating memorable and strong passwords," *Comput. Secur.*, vol. 85, pp. 423–435, Aug. 2019.

[14] A. Forget, S. Chiasson, P. C. van Oorschot, and R. Biddle, "Improving text passwords through persuasion," in *Proc. 4th Symp. Usable Privacy Secur.*, 2008, pp. 1–12.

[15] J. H. Huh, S. Oh, H. Kim, K. Beznosov, A. Mohan, and S. R. Rajagopalan, "Surpass: System-initiated user-replaceable passwords," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 170–181.

[16] A. Das *et al.*, "The tangled Web of password reuse," in *Proc. Internet Soc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, Feb. 2014.

[17] D. Schweitzer, J. Boleng, C. Hughes, and L. Murphy, "Visualizing keyboard pattern passwords," in *Proc. 6th Int. Workshop Vis. Cyber Secur. (VizSec)*, Oct. 2009, pp. 69–73.

[18] H.-C. Chou, H.-C. Lee, C.-W. Hsueh, and F.-P. Lai, "Password cracking based on special keyboard patterns," *Int. J. Innov. Comput., Inf. Control*, vol. 8, no. 1, pp. 387–402, 2012.

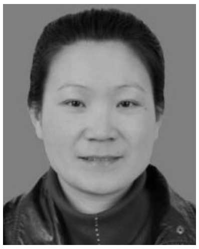
[19] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *Proc. 30th IEEE Symp. Secur. Privacy*, Piscataway, NJ, USA, Mar. 2009, pp. 391–405.

[20] S. Houshmand, S. Aggarwal, and R. Flood, "Next gen PCFG password cracking," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1776–1791, Aug. 2015.

[21] A. Borodin, F. E. Fich, F. M. A. Der Heide, E. Upfal, and A. Wigderson, "A tradeoff between search and update time for the implicit dictionary problem," *Theor. Comput. Sci.*, vol. 58, nos. 1–3, pp. 57–68, 1988.

[22] D. Florêncio, C. Herley, and P. C. Van Oorschot, "An administrator's guide to Internet password research," in *Proc. USENIX Conf. Large Installation Syst. Admin.*, 2014, pp. 35–52.

[23] M. Dell'Amico and M. Filippone, "Monte Carlo strength evaluation: Fast and reliable password checking," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, 2015, pp. 158–169.



JIANHUA SONG was born in Xiangyang, Hubei, China, in 1973. She is currently an Associate Professor with Hubei University. She is also a member of the Engineering and Technical Research Center of Hubei Province in Educational Informatization, the Engineering and Technical Research Center of Hubei Province in Software Engineering, and the Engineering Research Center of Hubei Province in Intelligent Government Affairs and Application of Artificial Intelligence. Her research interests include information security and network security.

DEGANG WANG was born in Ankang, Shaanxi, China, in 1996. He is currently pursuing the bachelor's degree with the School of Computer Science and Information Engineering, Hubei University, Wuhan, China. His research interests include password security and information security.

ZHONGYUE YUN was born in Jingshan, Hubei, China, in 1998. He is currently pursuing the bachelor's degree in information security with the School of Computer Science and Information Engineering, Hubei University, Wuhan, China.

XIAO HAN was born in Suizhou, Hubei, China. He is currently pursuing the bachelor's degree in information security with the School of Computer Science and Information Engineering, Hubei University, Wuhan, China.