Hindawi
Security and Communication Networks
Volume 2019, Article ID 5184643, 10 pages
https://doi.org/10.1155/2019/5184643

Research Article

An Explainable Password Strength Meter Addon via Textual Pattern Recognition

Ming Xu [1,2] **and Weili Han** [1,2]

[1] *Software School, Fudan University, China*
[2] *Shanghai Key Laboratory of Data Science, Fudan University, China*

Correspondence should be addressed to Weili Han; wlhan@fudan.edu.cn

Received 13 September 2018; Revised 5 November 2018; Accepted 20 December 2018; Published 13 January 2019

Academic Editor: Clemente Galdi

Textual passwords are still dominating the authentication of remote file sharing and website logins, although researchers recently showed several vulnerabilities about this authentication mechanism. When a user creates or changes a password, a website usually leverages a password strength meter (PSM for short) to show the strength of the password. When the password is evaluated as a weak one, the user may replace the password with a stronger or securer one. However, the user is usually confused when the password, especially a frequently used password, is shown as a weak one. We argue that an explainable password strength meter addon, which could show the reasons of *weak*, may help users to more effectively create a secure password. Unfortunately, we find few sites in Alexa global top 100 showing these details. Motivated to help users with an explainable PSM, this paper proposes an addon to PSMs providing feedbacks in the form of pattern passwords explaining why a password is weak. This PSM addon can detect twelve types of patterns, which cover a very large proportion among 70 million of leaked real passwords from high-profile websites. According to our evaluation and user study, our PSM addon, which leverages textual pattern passwords, can effectively detect these popular patterns and effectively help users create securer passwords.

1. Introduction

Although graphical passwords or other alternatives to textual passwords have been proposed in recent decades, textual passwords are still one of the most widespread methods of authentication in the Internet because of its convenient simplicity and sound implementation. However, researches reported some vulnerabilities about this authentication mechanism. Current works [1–3] show that users intend to choose weak passwords, which are usually easy to be remembered but vulnerable to be guessed. Meanwhile, study [4] reveals that textual passwords are often reused, which is also security threat among textual passwords.

To prevent users from generating weak passwords, system administrators usually leverage a variety of measures, including strict password composition policies [5] as well as password strength meters (PSMs for short). Password composition policies mean that a created password must meet certain constraints, such as at least eight characters in length, at least from two or three character classes, and including special characters. However, a strict password composition policy may make users confused. A user may generate a *satisfied* password, *e.g.*, `wanglei19951231`, in a direct way, where the user slightly adjusts the password to meet the prescribed policies. Yet the password is still weak since it inadvertently includes popular password patterns.

Another way to encourage users to select strong passwords is to employ PSMs. It is a good news in password research that a well-designed PSM does help guide end users to a securer password against online and offline password cracking [6]. However, the current PSMs often show confusing results. As reported in previous works [7], some PSMs often overestimate insecure passwords or underestimate secure passwords. For example, they would label the password of *haorenyishengpingan* as *strong* according to a full length. But it should be *weak*, because (1) it is composed of Chinese Pinyin with six Chinese characters, and (2) it is a very popular sentence in Chinese and often included by attackers

in an attack dictionary. In addition, the different PSMs employed by high-profile websites give highly inconsistent outcomes for same passwords [8]. On the other hand, current PSMs show colored bars alone without detail information. That is, they do not explain why a created password is weak. As a result, these weaknesses and inconsistencies might make users confused when users create or modify their original passwords.

Users widely leverage patterns to create their own passwords. These patterns can help users to remember created passwords, which are easy to be guessed unfortunately. Previous work [9] showed that an attacker who leverages regional patterns in China could result in an improvement of efficiency by 34% during guessing Chinese passwords. These results reveal pattern passwords could reduce textual passwords' security strength. Thus it is necessary to show users detailed patterns to guide the users to generate securer passwords.

On the other hand, we will show in Section 4 that these pattern passwords are so common that users may unconsciously apply these patterns to create their passwords. This paper leverages over 70 million leaked and publicly available passwords from five high-profile Chinese websites (CSDN [10], Tianya [11], Duduniu [12], 7k7k [13], 178.com [14]). We calculated each patterned password's proportion among these 70 million of leaked real password datasets and founded that the result is up to 40% where a password is composed of pure digits.

To help users choose stronger or securer passwords and promote users' security awareness, we argue that a PSM addon, which integrates with current PSMs and tells users their passwords are weak when their passwords follow some popular patterns, is useful. That is, our meter addon makes users understand that using pattern passwords are vulnerable to dictionary attacks. In addition, although there are three websites whose PSMs use a mechanism for detecting pattern passwords according to our survey, our PSM addon covers a larger number of patterns. That is, our PSM addon could detect as many weak pattern passwords as possible to achieve a higher accuracy for showing more weak pattern passwords than the existing three PSMs. Notably, our PSM addon firstly detects patterns oriented for Chinese, such as Pinyin patterns, which cover a large percentage among 70 million of leaked passwords.

We argue that our addon promotes development of PSMs for two reasons. First, our addon could integrate with current explainable meters [15] to provide more information. Second, our addon makes weak passwords understandable for users. Meanwhile, the main function of PSMs is to nudge weak passwords to stronger or securer ones for users. So our PSM addon does not disturb users with other passwords, such as medium, strong passwords, for better usability. Our PSM addon provides client side interfaces that give users instant feedback in web pages. There are two versions of our PSM addon, the private version could hide the details of our measurement and the public version which show the detail of our measurement. We also randomly invite 50 colleges to evaluate our PSM addon's effectiveness. The result shows their created passwords are in an improved strength.

The main contributions of this paper are as follows:

(1) We propose an explainable addon for PSMs with two improved features. One is that our PSM addon shows which patterns a password consists of. The other is that our PSM addon only reminds users with really weak passwords, which means we only show *real dangerous* patterns such as *wanglei19951231* or *wanglei13512341111* nor the *wanglei1995123113512341111* because the front of pattern has a small search space for attackers.

(2) We conduct a survey on PSMs employed by high-profile websites ranked in Alexa top 100. We find that there is few meters explaining why a password is weak. In addition, we calculated the proportion of these pattern passwords, which are mentioned in this paper, among 70 million leaked passwords. The results reveal that the proportion of pattern passwords is surprisingly large.

(3) We conduct a user study to investigate whether our PSM addon could help them nudge their original weak password, whether our PSM addon could improve their security awareness. The survey results show that our PSM addon is helpful since participants admit their generating password behaviors often fall into these patterns and they are willing to change their passwords to stronger or securer ones as the PSM addon becomes more understandable.

We begin by investigating previous research in the next section. Then, we provide details of our explainable PSM addon and discuss its algorithm in Section 3. Afterward, we present the evaluation, as well as the results of our proportion analysis of pattern passwords in Section 4. Then, we provide a user study and its analysis in Section 5. Finally, we conclude the paper and show our future works in Section 6.

2. Related Work

As is known to us, estimating the strength of passwords is something what we called *proactive password checkers or password strength meters (PSMs)*. They are generally represented as a colored bar indicating a weak password by a short green bar or a strong password by a red bar, accompanied by a word qualifying password strength, *e.g.*, weak, medium, or strong. According to prior studies [16, 17], these meters are usually based on password length or the character classes used. But they frequently show wrong strength of passwords [16].

*PSM*s have been around for decades. NIST proposed a method to measure the password strength based on password's entropy (or Shannon's entropy) [18], which relies on purely statical methods. To illustrate, it evaluates the password strength by a math model based on password length and selectable alphabet size. Unfortunately, current studies [7] showed this method is only suitable for evaluating the strength of randomly generated passwords rather than user-chosen passwords.

Later, researchers proposed new methods, such as password cracking methods [19], other than statistics to measure password strength. The principle behind them is to model common behaviors in human creating passwords. For instance, a common dictionary and various mangling rules are used to generate guesses. Although they produce fairly accurate guesses [20], the dictionary space and the amount of mangling rules required is too large to provide real-time measure results on the client side. Compared to JtR [21], PCFG-based model can crack 128%-129% of passwords at the same guess times. In order to better measure the password strength, Kelly *et al.* [22] proposed to use the number of cracking password to measure the strength of password from the perspective of the attacker.

Password guessing attracted a lot of attention in the academic field. First, Narayanan *et al.* [23] discussed a password guessing algorithm based on Markov models. The principle behind the model is based on the frequency of each character. The main advantage of Markov-based methods is that it can make an accurate assessment of the strength of a password that never occurred in training set. However, evidences [24] showed that these methods assign very high scores to slightly altered weak password. Moreover, it might confuse users about password strength by the same score between a slightly altered password *passwo1rd* and a random sequence "jdgsa234". Second, Weir *et al.* [25] proposed another password guessing algorithm based on PCFG method for the first time. The PCFG-based guessing method, which is abbreviated from *probabilistic context-free grammar*, uses a probabilistic model to model the password structure from a large scale training set and improves the efficiency of password guessing. Third, William *et al.* [26] proposed using artificial neural networks to model password cracking, which made a huge progress since his model is so small that powers instant result and the method is relatively accurate.

Notably, Golla *et al.* [27] demonstrated currently used measures to determine accuracy of strength meters are not precise and proposed a set of properties that a meter needs to fulfill. However, strength accuracy is not our focus. The above researchers are all focus on accuracy or algorithms behind PSMs. Our novelty lies on an explainable addon for PSMs, which adds explanations with pattern passwords to make users understand why their original passwords are weak.

All of the above measure methods are focusing on how to optimize the underlying password evaluation and cracking algorithms [28]. Yet, *zxcvbn* [29], Dropbox's PSM, is a novel password strength meter currently, which added a reducing score mechanism based on weak password patterns. The following patterns are considered: repeat (*e.g.*, sss, sdsdsd, and 1asd1asd); sequence (*e.g.*, 123; efghj); keyboard (*e.g.*, qwerty); date (*e.g.*, 5/6/1991; 07081994). If any of these weak patterns appear in a measured password, the entropy value is reduced accordingly. But the patterns *zxcvbn* detected are not enough to show the reasons why a measured password is weak. Our PSM addon presented in this paper may show more patterns than ones from *zxcvbn*. That is, our meter can detect regional patterns [9], phone number, and date's six variants.

Similarly, Ur *et al.* [15] proposed a data-driven password meter, which combines neural networks and numerous heuristics to score passwords and generate data-driven text feedback about user's password. The meter has a user-friendly design to provide feedback mainly from the following three levels: (1) suggestions to avoid dictionary words and keyboard patterns; (2) moving uppercase letters and digits away from the front or the end of a password; (3) including digits and symbols. The meter shows two kinds of pattern passwords to users, whereas our PSM addon leverages enough common pattern passwords as many as possible. That is, our PSM addon has superiority over Ur's meter in the level of detecting popular pattern passwords. In addition, Ur's meter might not be suitable for users in different regions, especially Chinese users since they can only show English dictionary words rather than Chinese Pinyin. Chinese users may tend to choose Pinyin instead of English words. We argue that it is necessary to target PSMs to Chinese users since Chinese account for a large proportion of netizens. Furthermore, our PSM addon could be integrated with other PSMs.

It is a challenging task to design a PSM addon that could run on the client side and could make users understandable about their original weak password. The current PSMs mostly focus on underlying algorithm but not study users' perception of *PSM*s, and few warning messages often result in users' confusion. We then propose an explainable addon for PSM, which could be integrated with other PSMs to provide more details with patterns. To the best of our knowledge, we are the first to show an addon with enough common patterns to remind users of pattern passwords' danger. From our users study, our explainable PSM addon can be observed that the changing password behavior has been significantly improved.

3. Explainable Password Strength Meter Addon

3.1. Overview. Here we propose an explainable PSM addon, which could detect popular pattern passwords as well as integrate with current PSMs, leading to current PSMs' enhancements that could point weak pattern passwords out to users. Our design will show twelve patterns. Specially, the twelve patterns include pure digits, which include telephone numbers, wire phone numbers, and dates; Pure letters, which include Chinese Pinyins; Two patterns combination including Pinyin+phone number patterns and Pinyin+date patterns; Special format including email addresses and keyboard patterns. In addition, we define twelve status codes to represent each type of pattern to unify the international standard. The match algorithms for these patterns are described in Section 3.2. There are two versions about our PSM addon: public version and private version. As is shown in Box 1, the public version does not hide passwords as black dot and can show more specific information to users, such as Pinyin's Chinese characters and zigzag keyboard patterns. In addition, as is shown in Box 2, the private version does hide passwords as black dot for privacy and hides a part of details to users. Both versions can be lightweightly deployed to web pages. We

{
“statusCode”: 402,
“pattern”: “Pinyin:hao,ren,yi,sheng,ping,an, 6 syllable”,
“proportion”: “Pinyin patterns account for approximately 5% of millions of passwords.”,
“hint”: “Pinyin patterns are dangerous. Avoiding Pinyin pattern passwords will be safer”
}

Box 1: The public version of our explainable PSM.

{
“statusCode”: 402,
“pattern”: “Pinyin pattern,6 syllable”,
“proportion”: “Pinyin patterns account for approximately 5% of millions of passwords.”,
“hint”: “Pinyin patterns are dangerous. Avoiding Pinyin pattern passwords will be safer”
}

Box 2: The private version of our explainable PSM.

look forward to improving the usability and security of PSMs by adding user's understandability.

3.2. Pattern Match Algorithms

3.2.1. Dates Pattern Matching Algorithm. According to Li's study et al. [30], six-digit dates can be classified into three formats: YYMMDD, MMDDYY, and DDMMYY. Similarly, eight-digit dates can be classified into YYYYMMDD, MMDDYYYY, and DDMMYYYY. Meanwhile, there may be false positive where a general six-digit number is considered as a date. The 30 date are, respectively, **111111, 123123, 111000, 112233, 100200, 100100, 111222, 121212, 520520, 110110, 123000, 111333, 101010, 110120, 102030, 110119, 121314, 010203, 122333, 121121, 101101, 521125, 321123, 110112, 112211, 111112, 120120, 520521, 110111, and 131211,** Thus, we also remove the above 30 digits to reduce false positive rate. The procedure to identify whether a password is composed of dates is shown in Algorithm 1.

3.2.2. Phone Number and Wire Phone Number Pattern Matching Algorithm. It seems common that many people choose their phone numbers as passwords for better memory. However, we all know that many personal information has been leaked such as address, phone number, ID number, and so on. It is relatively easy for attackers to collect enough phone number datasets and make a dictionary attack to guess user's passwords. Meanwhile, study [31] shows a large number of personal information increase password guessing effectiveness. Thus we make our PSM addon to identify whether a password is composed of phone numbers or wire phone numbers. To achieve this purpose, we use regular expression methods, which have significant performance advantages over equivalent string processing, to match phone numbers or wire phone numbers. We expect that this warning message could make users avoid phone and wire phone numbers when creating passwords.

3.2.3. Pinyin Pattern Matching Algorithm. Chinese users are usually familiar with Pinyin to input Chinese characters to the cyberspace. The passwords with Pinyin patterns are vulnerable to dictionary attacks. Thus, it is of significance to show Pinyin patterns to users. In this algorithm, we adapt trie (or prefix tree) to improve Pinyin matching efficiency. We construct trie by inserting Chinese Pinyins one by one. Each trie node is composed of characters of each word. This algorithm uses common prefix of each string to overhead of query time for purpose of improving efficiency. Meanwhile, we could not only match Pinyin patterns, but also show that syllables that make up the selected Pinyin by users. Showing Pinyin patterns and its composition details would undoubtly enhance users' understandability about PSMs. We look forward to improving password safety awareness of users, expecially Chinese users.

3.2.4. Keyboard Pattern Matching Algorithm. There are passwords that look like randomness, such as *zxcvbn*, *qazsedc*, *1qaz2wsx*, and so on. However, it is a common combination named *keyboard pattern*. We drive this conclusion that keyboard patterns are very common in the total Chinese passwords from the prior research [9]. This result is consistent with our subsequent pattern measurement for proportion. We, respectively, divide these keyboard patterns into two categories: the same row (*e.g.*, qwertyuio); the zigzag type (*e.g.*, qazsedcft). We made a script to generate a dictionary and retrieve each pair of letters in the generated dictionary. That is, we adopt the idea of space for time. Out algorithm can match all input characters of keyboard. The procedure to identify whether a password is composed of keyboard pattern is shown in Algorithm 2.

3.2.5. Email, Pure Letter, and Pure Digit Pattern Matching Algorithm. In some password leakage incidents, the associated usernames and emails are also leaked along with the passwords. Intuitively, the username and email information

```
Input: S:a string.
Output: TRUE or FALSE.
(1) Define a Set D,a bad set T,D = NULL,T = the selected 30 numbers
(2) Define get-day[ ] = [31,29(if leap year) 28(else), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
(3) if S is not digit then
(4)    return FALSE
(5) else if S.length != 6 or 8 or S ⊆ T then
(6)    return FALSE
(7) end if
(8) for month = 0;month < 12; do
(9)    for day = 1;day < get-day[month]; do
(10)      D.add(year,month,day)
(11)   end for
(12) end for
(13) if S ⊆ D then
(14)   return TRUE
(15) else
(16)   return FALSE
(17) end if
```

Algorithm 1: Dates matching.

```
Input: S:a string.
Output: TRUE or FALSE.
(1) initialize:dict:for ever keyboard i,we map j that is adjacent to i. e.g.we map <A,Z> <A,Q> <A,S>
<A,W> <A,X> for A.
(2) if S.length<4 then
(3)    return FALSE
(4) end if
(5) count = 0,string S1,string S2
(6) for i = 1;i < S.length();i + + do
(7)    S1 = S[i]+S[i+1];S2 = S[i+1]+S[i]
(8)    if S1 ⊆ dict or S2 ⊆ dict then
(9)      count++
(10)     if count == S.length-1 then
(11)      return TRUE
(12)     else
(13)      return FALSE
(14)     end if
(15)   end if
(16) end for
```

Algorithm 2: Keyboard patterns matching.

have impacts on password security since people may follow the same or similar style in choosing their email as passwords. According to prior experiments [32], email based cracking achieves better performance with less guesses. So it is of significance that matching email and point email information out to users could be helpful for improving password security. As for pure letter and pure digit pattern, obviously the violence search space is too small to resist password cracking; thus it is worth matching.

As for technology for achieving this function, we still and use regular expression to match format of email pattern, the pure letter pattern, and the pure digit pattern.

3.2.6. Two-Pattern Combination Matching Algorithm. In our work, we are still matching two-pattern combinations since two combined patterns cannot defend against violent search attacks. The two-pattern combinations are, respectively, Pinyin+Date, Pinyin+Phone, and wire phone number and the last the pure letter+digit. If the input password does not match any single pattern, the input password falls into two-pattern matching program. We did not match the three-pattern combinations because they are complicated enough to withstand attacks; thus our PSM addon does not remind users with indifferent warnings. We believe this design will not bring unnecessary burden to users.

Table 1: Basic information of leaked passwords of the websites that are analysed in this paper.

	language	Site Address	Amount
CSDN	Chinese	http://www.csdn.net/	6,425,720
Tianya	Chinese	http://www.tianya.cn/	30,180,739
Duduniu	Chinese	http://www.duduniu.cn/	16,277,247
7k7k	Chinese	http://www.7k7k.com/	18,591,784
178.com	Chinese	http://www.178.com/	9,071,979
Total			**74,121,749**

4. Evaluation

In this section, we prove that our work is of significance from the following two parts. On the one hand, we found that only three PSMs among 100 meters that employed by Alexa's global top 100 websites explains the national of password strength or gives constructive advice when users created their password. Thus users are not willing to change the weak password due to confusion. On the other hand, we make an investigation about the proportion of each pattern among the leaked password sets. The results show that these patterns are so common among passwords that pattern matching is of significance to regulate user's password setting behavior.

4.1. Empirical Study Analysis of High-Profile Websites' PSMs. Due to high popularity of textual PSM among current high-profile websites, many prior research [6, 8] revealed its effectiveness as well as usability. As revealed in [8], there is little rational under their choices of policies imposed by many high-profile websites. For instance, the password *p@ssword* which is labeled *Medium* by Gmail may be labeled *strong* by Yahoo. This phenomenon may undermine users' trust in security advice. Evidence [6] showed a comprehensive the *prior quo* of PSM and concluded that a well-designed *PSM* does guide users towards a more secure password in an important account. However, the *status quo* of the PSM has changed how much in recent years remains an open question. So we make a large scale empirical study and analysis of PSMs employed by Alexa's global top 100 websites listed based on their traffic ranking (http://www.alexa.com/topsites). Note that there are some companies (*e.g.*, Google) which may offer various services such as email, search, and news and have a few affiliated sites among many countries. Fortunately, they generally rely on the same PSMs. So we consider all the affiliated sites as one.

Results. We performed a large scale empirical study and analysis about PSMs in two aspects. First, we want to have a general idea of difference of PSMs between current high-profile websites and ours. Table 3 shows the difference between *PSM*s of Alexa's global ranking top 10 websites and ours. Surprisingly, we found that up to three sites in top 10 sites give no hint to users. This phenomenon would greatly weaken the security of website's access control. The websites left are all giving the *weak* or *strong* warning messages based the underlying algorithm behind the PSMs. Sadly, most users do not understand the warning message due to technological or other reasons, so cannot nudge their original and weak password. Secondly, we calculated proportion of three most commonly used patterns detected by Alexa's top 100 websites. Table 2 shows our results. We draw the following conclusions:

Table 2: Proportion of three commonly used patterns matching scheme employed by Alexa's top 100 web sites.

Pattern	Percentage
Pinyin	0(0/100)
Date	3%(3/100)
Keyboard	2%(2/100)

(i) Only three websites (Google [33], Paypal [34], and Bet365 [35]) give users warning messages when users follow the date pattern to create passwords. Only Google and Paypay limit the use of keyboard patterns. In contrast, our explainable PSM addon is different from the above meters in that ours can match more formats of dates and keyboard patterns, such as the zigzag keyboard pattern (e.g., *qazsedc*), six kind of date format (e.g., *9512231, 123195, 311295*), and so on. Most importantly, Pinyin patterns are not included in the above meters. We argue that Pinyin patterns cannot be ignored since Chinese Pinyin plays an important role in increasing effectiveness when guessing Chinese passwords [9].

(ii) Most websites do not provide any explanations for their design choices, sometimes making them as a black box. We believe an explainable PSM addon could become a mainstream application industrially in the near future due to understandability.

4.2. Measuring the Proportion of These Popular Patterns among Leaked Passwords

4.2.1. Dataset Statement. We analyzed a corpus of over 70 million passwords from multiple famous websites including CSDN, Tianya, 7k7k, 178, and Duduniu, respectively. All the leaked passwords are publicly available for downloading. We hereby declare that we never utilized these leaked data for any other purpose rather than research.

An unfortunate incident, where passwords from five famous websites, including CSDN, Tianya, Duduniu, 7k7k, and 178, were leaked in several consecutive days, happened in the end of 2011. The total number of leaked accounts is over 70 million, and all the leaked passwords are in plaintext. We leverage these leaked and publicly available passwords, as summarized in Table 1.

Table 3: A table to show the difference between PSMs of Alexa top 10 sites and ours.

	haorenyishengpingan	19951231	qwertyuiop
our explainable meter	StatusCode:402 Pattern:Pinyin Proportion:up to 5.4%	StatusCode:302 Pattern: Dates Proportion:up to 12.9%	StatusCode:201 Pattern:Keyboard proportion:up to 10.3%
Google	Password strength: Strong	Password strength: Strong	Password strength: Weak
Facebook	no hint	no hint	no hint
Baidu	6~14 characters in length Support for numbers, uppercase and lowercase letters and punctuation No spaces allowed	6~14 characters in length Support for numbers, uppercase and lowercase letters and punctuation No spaces allowed	6~14 characters in length Support for numbers, uppercase and lowercase letters and punctuation No spaces allowed
Reddit	Weak password strength	Weak password strength	Weak password strength
Yahoo	no hint	no hint	no hint
Tencent	Must contain at least 2 of letters, numbers, and symbols	Must contain at least 2 of letters, numbers, and symbols	Must contain at least 2 of letters, numbers, and symbols
Amazon	no hint	no hint	no hint
Taobao	strength: medium	the password does not meet requirements	medium:weak
Twitter			

Table 4: The result dataset of evaluating each pattern proportion among leaked password set.

	CSDN	7k7k	178	Duduniu	Tianya
Pinyin	5.04%	5.11%	5.42%	3.87%	4.25%
Keyboard	10.31%	7.81%	9.82%	3.81%	10.07%
Phone number	2.95%	1.97%	3.77%	2.49%	2.26%
Wire phone number	1.27%	0.00%	0.54%	0.29%	0.57%
Date	9.57%	10.27%	4.67%	3.34%	12.86%
Email	0.07%	0.02%	0.01%	0.10%	0.02%
Pure digit	22.56%	40.83%	32.06%	21.71%	39.44%
Pure letter	7.37%	5.79%	3.19%	7.56%	5.66%
Pinyin+phone/wire phone number	0.50%	0.17%	0.11%	0.06%	0.43%
Pinyin+date	5.82%	3.28%	4.57%	5.45%	3.30%
Digit+letter	34.42%	24.74%	35.81%	51.27%	2.10%

4.2.2. Measuring Methods. After we download the original database file, we write a script to extract the password-only file from it. Then we read the file to count each password pattern using the node.js framework. Our result is showed in Figure 1, as summarized in Table 4. From Table 4 and Figure 1, we can obtain following observations:

(i) It is not surprising that the proportion of pure digits and digits+letters patterns account for the most. But keyboard patterns as well as dates patterns' proportion are as much as 8% among 70 million of passwords. These results show that pattern passwords are so common that we cannot ignore them.

(ii) Pinyin and Pinyin+digit patterns occupy about 4%, and phone number occupies about 2%, followed by Pinyin. Email pattern accounts for the smallest proportion.

5. User Study

To better evaluate the effectiveness of our explainable PSM addon, we conducted a questionnaire based on password-generating questions, recruiting participants among collage students. Participants, whose ages were at least 18 years, were 50 colleges with technical background. Participants were informed that the main function of the questionnaire was to investigate the effectiveness of explainable PSM addon, which can be downloaded from attached file for trying out the PSM addon. After that, participants were assigned with four questions.

We conducted this questionnaire from the following perspectives. First, we asked them *Do passwords you normally create include these pattern passwords?* We want to know whether these participants will fall into the bad habits of pattern passwords and the most commonly used pattern passwords, even if these participants are all with technical

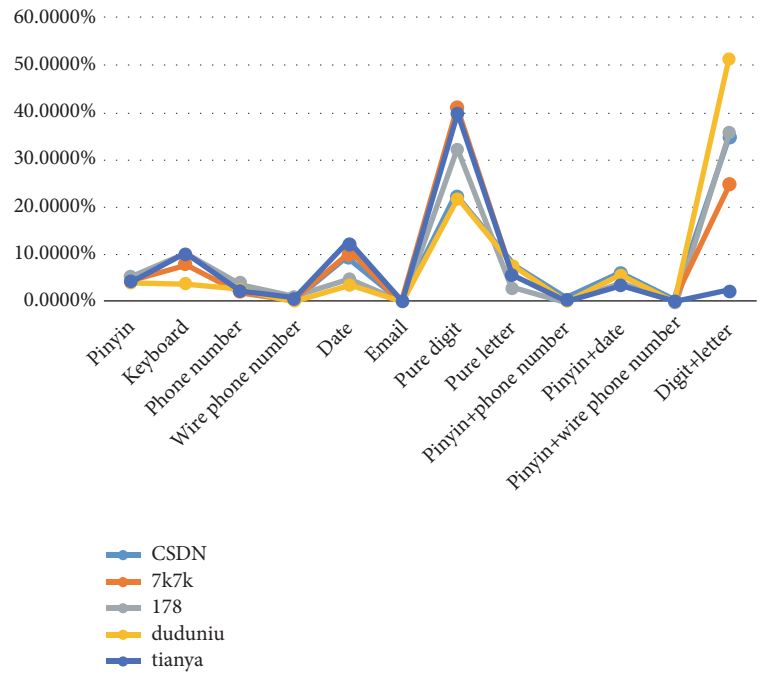

Figure 1: The proportion of pattern passwords among 70 million leaked passwords.

background. Second, we asked another question. The question is *A PSM addon gives a warning message and shows your password consists of commonly used patterns and is vulnerable to attack when you create original password. Would you like to change it?* We believe that this behavior, changing weak password, is one of measurements of a well-designed PSM. So we want to know whether our explainable PSM addon helps users towards a stronger or securer password, thus regulating their password-generating behaviors. Third, our question is *Does your password security awareness be improved under such PSM addon?* We want to have an intuitive understanding of whether our PSM addon helps improve their security awareness after trying out our explainable PSM addon. Participants are expected to give a referable opinion since they all have technological background. Last, we want to know if there are other password patterns used by users but not covered by this paper. So we asked *Do you have other commonly used password patterns when creating passwords other than those mentioned in this paper? Do they will choose any pattern passwords when generating passwords after trying out our PSM addon?* Then we collected the answers of these participants and reached results and corresponding analysis.

Result. For the first question, Table 5 lists the top five most popular pattern passwords that users have chosen from these 12 pattern passwords mentioned in this paper. From this table, we can see that the most popular pattern password is digit+letter, and Pinyin and phone number pattern follow. It is not surprising that the most popular pattern is digit+letter combination since it contains several small patterns. For small patterns, Pinyin pattern and phone number are most commonly used two patterns. This result may be because the questionnaire is implemented by Chinese college. They are familiar with Chinese Pinyin and can remember a Pinyin pattern password easily. We can see that although they all have a certain degree of technological background, they are still falling into a bad habit of pattern passwords when creating password. This phenomenon reminds us and websites' managers that pattern passwords cannot be ignored. For the second question, this investigation showed that 29 out of 50 people said that they will change their original password under this PSM addon's warning information. We reckon the reason may be users might have motivation to change their behaviors when warning messages become targeted and understandable by PSMs, resulting in improved password-generating behaviors. For the third question, 35 out of 50 participants said that our explainable PSM really helps them enhance their password security awareness. These responses reveal that users tend to accept understandable messages given by explainable PSMs. The reason may be that our PSM addon tells users that pattern passwords are dangerous; thus users will change this password-generating behavior. Last, we want to know if there are other patterns that users will use and then do they choose pattern passwords when generating passwords. Results show that *Identification number*, *QQ number*, *Bank card number*, and *English words*

Table 5: Top five most popular pattern passwords and their proportions selected by participants.

	Pattern	Percentage
1	Digit+letter	82%(41/50)
2	Pinyin	62%(31/50)
3	Phone number	58%(29/50)
4	Pure digit	54%(27/50)
5	Pure letter	34%(22/50)

are all used pattern passwords by participant. Even if there exist some other patterns, most of them admit that they will regularize their choosing pattern passwords behavior after trying this PSM. That is, out explainable PSM addon will make users understand the danger of pattern passwords; thus they will avoid these unsafe passwords, which show that our PSMs could effectively help users.

6. Conclusion and Future Work

It is an important purpose of PSMs to nudge users to change a weak password to a stronger or securer one. Our study show that providing users with patterns in passwords could remind users the danger of pattern passwords. To our knowledge, our PSM addon could improve PSM's understandability and usability according to our user study. In this paper, we found only three websites (Google, Paypal, and Bet365) among Alexa top 100 websites provide users with small number of pattern passwords. However, our PSM addon, which may integrate with current PSMs, achieves more accuracy than these 3 meters in detecting more patterns. That is, Our PSM addon could show weaker passwords. Furthermore, we calculated the proportion of these pattern passwords among 70 million leaked passwords. Then we found that passwords with keyboard patterns account for up to 10.3%. These results further confirmed that users often choose pattern passwords. Treating password strength estimation as a black box always weakens the purpose of current PSMs. Our PSM addon could make current PSMs work better. In addition, we design and implement the PSM addon in client side, which is convenient to be used by websites and to be integrated with current PSMs. Finally, we found that users intend to change their original passwords and avoid these pattern passwords with help of our explainable PSM addon according to our user study.

Further research may involve more popular patterns, such as *QQ number*. Meanwhile, the number of our user study is relatively small. It is limited to college student population. A user study with more people and more professional distribution deserve further investigation. In addition, our findings have implications for pattern recognition to enhance meter's effectiveness. This result could provide a new direction for the research of PSMs, that is, to find out potential relationship between human-computer interaction and usable security among Internet users.

Data Availability

We analyzed a corpus of over 70 million passwords from multiple famous websites including CSDN, Tianya, 7k7k, 178, and Duduniu, respectively. All the leaked passwords are publicly available for downloading. We hereby declare that we never utilized these leaked data for other purposes rather than research.

Disclosure

Weili Han is the corresponding author of the paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by NSFC (Grants No.: U1836207 and 61572136), National Key R&D Program of China (Grant No.: 2018YFC0830900), and STCSM (Grant No.: 18511103600).

References

[1] J. Yan, A. F. Blackwell, R. J. Anderson, and A. Grant, "Password memorability and security: empirical results," *IEEE Security & Privacy*, vol. 2, no. 5, pp. 25–31, 2004.

[2] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in *Proceedings of the NDSS*, 2014.

[3] M. Bishop and D. V. Klein, "Improving system security via proactive password checking," *Computers & Security*, vol. 14, no. 3, pp. 233–249, 1995.

[4] W. Han, Z. Li, M. Ni, G. Gu, and W. Xu, "Shadow attacks based on password reuses: A quantitative empirical view," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 2, pp. 309–320, 2018.

[5] P. G. Inglesant and M. A. Sasse, "The true cost of unusable password policies: password use in the wild," in *Proceedings of the CHI*, 2010.

[6] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, "Does my password go up to eleven? the impact of password meters on password selection," 2013.

[7] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proceedings of the ACM Conference on Computer and Communications Security*, E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, Eds., pp. 162–175, ACM, 2010.

[8] D. Wang and P. Wang, *The Emperor's New Password Creation Policies*, Springer International Publishing, Cham, Switzerland, 2015.

[9] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis of chinese web passwords," in *Proceedings of the 23rd USENIX Security Symposium*, K. Fu and J. Jung, Eds., pp. 559–574, USENIX Association, San Diego, CA, USA, August 20-22, 2014.

[10] "Csdn," http://www.csdn.net/company/about.html.

[11] "Tianya," http://help.tianya.cn/about/history/2011/06/02/166666.shtml.

[12] "Duduniu," http://baike.baidu.com/view/1557125.htm.

[13] "7k7k," http://www.7k7k.com/html/about.htm.

[14] "178.com," http://www.178.com/s/information/about.html.

[15] B. Ur, F. Alfieri, M. Aung et al., "Design and evaluation of a data-driven password meter," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems CHI '17*, pp. 3775–3786, ACM, New York, NY, USA, 2017.

[16] X. de Carné de Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," 2014.

[17] B. Ur, P. Gage Kelley, S. Komanduri et al., "How does your password measure up? the effect of strength meters on password creation," *Usenix Security Symposium*, 2012.

[18] Nist and E. Aroms, "Nist special publication 800-63 electronic authentication guideline," 2012.

[19] D. Goodin, "Anatomy of a hack: How crackers ransack passwords like qeadzcwrsfxv1331," Ars Technica, 2013, http://arstechnica.com/security/2013/05/how-crackers-make-minced-meat-out-of-yourpasswords/.

[20] B. Ur, S. M. Segreti, L. Bauer et al., "Measuring real-world accuracies and biases in modeling password guessability," in *Proceedings of the 24th USENIX Security Symposium, USENIX Security 15*, J. Jung and T. Holz, Eds., pp. 463–481, USENIX Association, Washington, DC, USA, 2015.

[21] "Wordlist from john the ripper," http://download.openwall.net/pub/passwords/wordlists/.

[22] P. G. Kelley, S. Komanduri, M. L. Mazurek, and R. Shay, *Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms*, 2012.

[23] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proceedings of the 12th ACM Conference on Computer and Communications Security, CCS 2005*, V. Atluri, C. A. Meadows, and A. Juels, Eds., pp. 364–372, ACM, Alexandria, VA, USA, 2005.

[24] H. Tupsamudre, V. Banahatti, and S. Lodha, "POSTER: improved markov strength meters for passwords," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds., pp. 1775–1777, ACM, Vienna, Austria, 2016.

[25] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *Proceedings of the 30th IEEE Symposium on Security and Privacy (S&P 2009)*, pp. 391–405, IEEE Computer Society, Oakland, Calif, USA, 2009.

[26] W. Melicher, B. Ur, S. Komanduri, L. Bauer, N. Christin, and L. F. Cranor, "Fast, lean, and accurate: Modeling password guessability using neural networks," in *Proceedings of the 2017 USENIX Annual Technical Conference, USENIX ATC 2017*, D. Da Silva and B. Ford, Eds., USENIX Association, SNTC, CA, USA, 2017.

[27] M. Golla and M. Dürmuth, "On the accuracy of password strength meters," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018*, D. Lie, M. Mannan, M. Backes, and X. F. Wang, Eds., pp. 1567–1582, ACM, Toronto, Canada, 2018.

[28] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *Proceedings of the 2014 IEEE Symposium on Security and Privacy, SP 2014*, pp. 689–704, IEEE Computer Society, Berkeley, Calif, USA, 2014.

[29] D. L. Wheeler, "zxcvbn: Low-budget password strength estimation," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pp. 157–173, USENIX Association, Austin, Tex, USA, 2016.

[30] W. Han, Z. Li, L. Yuan, and W. Xu, "Regional Patterns and Vulnerability Analysis of Chinese Web Passwords," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 258–272, 2016.

[31] Y. Li, H. Wang, and K. Sun, "A study of personal information in human-chosen passwords and its security implications," in *Proceedings of the 35th Annual IEEE International Conference on Computer Communications, INFOCOM 2016*, pp. 1–9, IEEE, SF, CA, USA, 2016.

[32] S. Ji, S. Yang, X. Hu, W. Han, Z. Li, and R. Beyah, "Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 550–564, 2017.

[33] "Google," http://www.google.cn.

[34] "Paypal," http://www.PayPal.com.

[35] "Bet365," http://www.bet365.com.