Understanding Passwords of Chinese Users: A Survey and Empirical Analysis

Ding Wang, Haibo Cheng, Ping Wang, Xinyi Huang, Chao-Hsien Chu Senior Member, IEEE

Abstract—While the computer security world has changed a lot over the last two decades, textual passwords remain the dominant authentication mechanism over the Internet and are likely to persist in the foreseeable future. Much attention (e.g., user surveys and empirical analysis) has been paid to passwords chosen by English users, yet relatively little is known about how non-English users select passwords. In this work, we conduct so far the first user survey on the password behaviors of Chinese users, revealing a number of users’ basic coping strategies for managing passwords when they are confronted with the demanding tasks of keeping track of many accounts and passwords.

We further perform an empirical analysis of 100 million Chinese web passwords in a comparison with 30 million English ones, a corpus among the largest ones ever studied. We identify a number of interesting structural and semantic characteristics in Chinese passwords, and also examine their security by employing two state-of-the-art password cracking techniques (i.e., probabilistic context-free grammars (PCFG) and Markov models). Particularly, our cracking results reveal a “reversal principle”: when the guess number allowed is small, Chinese passwords are much weaker than their English counterparts, yet this relationship will be reversed when the guess number is large. This well reconciles two conflicting claims about the strength of Chinese web passwords made by Bonneau in 2012 and Li et al. in 2014, respectively. At 10^3 guesses, the success rate of our improved PCFG-based attack against the Chinese datasets is from 33.2% to 49.8%, indicating that our attack can crack 92% to 188% more passwords than the best record reported by Li et al. in 2014. We also discuss the implications of our findings. This work is expected to help facilitate both security administrators and users to gain a better understanding of the vulnerability of Chinese passwords, as well as to shed light on future password research.

Index Terms– User authentication, Password security, Semantic pattern, Probabilistic context-free grammar, Markov model.

I. INTRODUCTION

Password-based authentication is the dominant form of access control in almost every web service today. Though the security pitfalls of textual passwords were revealed as early as four decades ago [1] and various alternative authentication methods (e.g., graphical passwords and multi-factor authentication [2]) have been proposed since then, they stubbornly survive and reproduce with every new web service while Internet technologies have advanced by leaps and bounds in other areas. For one reason, textual passwords offer advantages, such as low deployment cost, easy recovery and remarkable simplicity, which cannot always be matched by their alternatives [3]. The matter is further complicated by the shortage of effective methods to quantify the obscure costs of replacing passwords [4], while marginal gains are often insufficient to reach the activation energy that is necessary to overcome the significant transition costs. Thus, passwords remain the primary method for user authentication and are likely to persist in the foreseeable future.

Despite its ubiquity, password authentication is stuck with an inherent tension [5]: truly random passwords are hard for users to memorize, while user-memorable passwords may be highly predictable. To deal with this notorious “security-usability” dilemma, researchers have devoted significant efforts [6]–[9] to the following two types of studies. Type-1 research aims at evaluating the strength of a password dataset by gauging its Shannon entropy as in NIST-800-63 [10], or by measuring its “guessability” [9]. The latter notion characterizes the fraction of passwords that, at a given number of guesses, can be cracked by cracking algorithms like the probabilistic context-free grammars (PCFG) [11] and Markov-based attacks [7].

Type-2 research attempts to reduce the use of weak passwords, and mainly two approaches have been utilized: proactive password checking [12] and password strength meter [13]. The former checks the user-selected passwords and only accepts the ones that comply with the system policy (e.g., at least 8 characters long). The latter is typically a visual feedback of password strength, often presented as a colored bar to help users create stronger passwords [14]. Most of today’s leading sites employ a combination of these two approaches to restrict users from choosing weak passwords. In this work, though we mainly focus on type-1 research, one can see that our results are also helpful for type-2 research.

Existing literature mainly focuses on passwords chosen by English users, or more precisely, by netizens using the Latin alphabet, yet little attention has been paid to the characteristics and strength of passwords generated by those who use other native languages. For instance, password “wangle123” is currently deemed “Strong” by many high-profile password strength meters (PSMs) like those of AOL, Google, IEEE, and Sina weibo. However, as shown in [15], this password is highly prone to guessing, for “wanglei1” is a very popular Chinese Pinyin (i.e., the Chinese phonetic alphabet) name but not a random string of length seven. Failing to catch this is equal to blindly overlooking the weaknesses of Chinese passwords, posing the corresponding account at high risks.

A. Motivations

To our knowledge, so far there has been no satisfactory answer to the following important questions: (i) What’s the password management behaviors of Chinese users (e.g., creation and memorization)? (ii) Are there structural or semantic characteristics that differentiate Chinese passwords from English ones? (iv) What’s the security strength of Chinese passwords? (iv) Are they weaker or stronger than English passwords? There have been 668 million Chinese netizens by Dec., 2015 [16], which account for over 25% (and also the largest fraction) of the world’s Internet population, and it is of great importance to answer these questions to provide both security engineers and Chinese users with necessary security guidance. For instance, if the answer to the second question is affirmative, then it highly indicates that the password policies...
A few password studies (e.g., [7]–[9]) have employed some Chinese datasets, yet they mainly deal with the effectiveness of various probabilistic cracking models, and relatively little attention is given to the differences between Chinese passwords and English passwords. Besides, to date most previous works (e.g., [7]–[9]) on Chinese passwords mainly employ an empirical approach. Yet, this approach is inherently unable to reveal many important user password behaviors, such as what fraction of users write down passwords, how many different passwords users have and what’s the sentiment (e.g., satisfactory or not) about their own current management of passwords.

As far as we know, Li et al.’s work [17] may be the closest to the current paper. However, they also only conduct an empirical analysis, and many important Chinese user behaviors in managing passwords are left untouched. Besides, a number of fundamental characteristics, such as top popular passwords, length distribution, frequency distribution and Pinyin-name-based semantic patterns, have not been explored in their work. What’s more, they proposed an improved PCFG-based cracking algorithm and at $10^{10}$ guesses, their best success rate is about 17.3%, while our improved PCFG-based algorithm can achieve success rates from 29.41% to 39.47% at just $10^7$ guesses. In addition, Li et al. invalidated the remarks made in [18] that Chinese passwords are among the strongest ones to guess than English passwords, and they concluded that “the strength of the passwords of Chinese and English users is similar”. Our results show that Li et al.’s conclusion is still biased. Last but not the least, we demonstrate that two of its five Chinese datasets are improperly pre-processed when performing data cleansing. Therefore, the effectiveness of the results reported in [17] might be impaired. For example, Li et al. reported that there are 3,639,517 (11.78%) passwords in Tianya with consecutively exactly eight digits, yet the actual value is 2.4 times larger: 8,664,708 (28.04%).

### B. Contributions

To answer the above four fundamental questions, we first conduct an online user survey on the password management behaviors of Chinese users. To complement our user survey, we further perform a large-scale empirical analysis by leveraging over 100 million real-world passwords from six popular Chinese sites and more than 30 million passwords from three English sites, one of the largest corpuses of user-chosen passwords ever studied. Benefiting from the plaintext form of these 130M passwords, we seek for fundamental properties of user-chosen passwords and, for the first time, manage to identify several distinct characteristics of Chinese web passwords in comparison to English ones. In summary, we make the key contributions:

- **An empirical analysis.** By leveraging 100 million real-life Chinese passwords, we, for the first time: (1) explore the name semantics in Chinese passwords, and show that there is an *every one in nine* probability that Chinese users insert their Pinyin names into passwords, while the fraction of English passwords that include an English name (with length $\geq 5$) reaches 24.3%, i.e., one in four; (2) reveal that *every one in six* Chinese users inserts a 6-digit date (e.g., birthdate) into their password; (3) provide a quantitative measurement of to what extent user passwords are influenced by their native language; (4) show that passwords chosen by these two distinct user groups are of quite similar frequency distributions.

- **An reversal principle.** We employ two state-of-the-art password-cracking algorithms (i.e., PCFG-based and Markov-based [7]) to measure the strength of Chinese web passwords. Based on the identified characteristics of Chinese passwords, we improve the PCFG-based algorithm to more accurately capture passwords that are of a monotonically long structure (e.g., “lqa2ws3ed”). At $10^7$ guesses, our algorithm can crack 92% to 188% more passwords than the results in [17]. Particularly, we reveal a “reversal principle”: when the guess number allowed is small, Chinese passwords are much weaker than their English counterparts, yet this relationship is reversed when the guess number is large, thereby reconciling the contradictory claims made in [17], [18].

- **Some insights.** We highlight some useful insights for password cracking, strength meters and creation policies. To our knowledge, we are the first to provide a *large-scale empirically grounded evidence* that supports for the hypothesis proposed in [14], [19]: users rationally choose more complex and secure passwords for accounts associated with higher value, and knowingly select weaker passwords for lower-value web service even though the latter imposes a stricter policy.

### II. RELATED WORK

In this section, we briefly review prior research on password characteristics and security to facilitate later discussions.

#### A. Password characteristics

**Basic statistics.** In 1979, Morris and Thompson [1] analyzed a corpus of 3,000 passwords and reported that 71.12% of passwords are no more than 6 characters long and 13.93% of passwords are non-alphanumeric characters. In 1990, Klein [12] collected 13,797 computer accounts from his friends and acquaintances around US and UK, and observed that users tend to choose passwords that can be easily derived from dictionary words: a dictionary of 62,727 words is able to crack 24.21% of the collected accounts and 51.70% of the cracked passwords are shorter than 6 characters long. In 2004, Yan et al. [5] found that user-chosen passwords are likely to be dictionary words since users have difficulty in memorizing random strings, and that the lengths of passwords in their study (involving 288 participants) are on average between 7 and 8.

In 2012, Bonneau [18] conducted a systematic analysis of 70 million Yahoo passwords. This work examined dozens of subpopulations based on demographic factors (e.g., age, gender and language) and site usage characteristics (e.g., email and retail), and found that “even seemingly distant language communities choose the same weak passwords”. Particularly, Chinese passwords are among the most difficult ones to crack. In 2014, however, Li et al. [17] argued that Bonneau’s dataset
is not representative of general Chinese users, because Yahoo users are familiar with English. Accordingly, Li et al. leveraged a corpus of five datasets from Chinese sites and observed that Chinese users like to use digits when creating passwords, as compared to English users who like to use letters to build passwords. However, as an elementary defect, two of their Chinese datasets have not been cleaned properly (see Section IV-B), which might lead to inaccurate measures and biased comparisons. More importantly, several critical password properties (such as the most popular passwords, the length and frequency distributions of passwords) remain to be explored.

In 2014, Ma et al. [7] investigated password characteristics about the length and the structure of six datasets, three of which are from Chinese websites. Nonetheless, this work mainly focuses on the effectiveness of probabilistic password cracking models and pays little attention to the deeper semantic patterns of passwords (e.g., no information is provided about the role of Pinyin, names or dates).

Semantic patterns. In 1989, Riddle et al. [20] found that birth dates, personal names, nicknames and celebrity names are popular in user-chosen passwords. In 2004, Brown et al. [21] confirmed this by conducting a thorough survey that involved 218 participants and 1,783 passwords, and they reported that the most frequent entity in passwords is the self (accounts for 66.5%), followed by relatives (7.0%), lovers and friends; also, names (32.0%) were found to be the most common information used, followed by dates (7.2%). In 2012, Veras et al. [22] examined the 32 million RockYou dataset by employing visualization techniques and observed that 15.26% of passwords contain sequences of 5-8 consecutive digits, 38% of which could be further classified as dates. They also found that repeated days/months and holidays are popular, and when non-digits are paired with dates, they are most commonly single-characters, or names of months.

In 2014, Li et al. [17] showed that Chinese users tend to insert Pinyin and dates into their passwords. However, many other important semantic patterns (e.g., Pinyin name and mobile number) are left unexplored. In addition, due to an imprudent processing of data cleansing and improper tuning of cracking algorithms (see Sec. V), Li et al.’s measurement of the strength of Chinese passwords will be shown to be biased. In 2015, Ji et al. [8] noted that user-IDs and emails have a great impact on password security. For instance, 53.24% of Dodonew passwords can be guessed by using user-IDs within, on average, 706 guesses. This motivates us to investigate to what extent the Pinyin names and Chinese-style dates impact the security of Chinese passwords.

B. Password security

A crucial research area in password cryptography is to study the security strength of user-chosen passwords. To avoid using the brute-force attack, earlier works (e.g., [12], [20]) use a combination of ad hoc dictionaries and mangling rules to model the common password generation practice and see whether user passwords can be successfully rebuilt in a period of time. This technique greatly improves the cracking efficiency and has given rise to automated tools such as John the Ripper (JTR).

Borrowing the idea of Shannon entropy, the NIST-800-63-2 guide [10] attempts to use the concept of password entropy for estimating the strength of password creation policy underlying a password system. Password entropy is calculated mainly according to the length of passwords and augmented with bonus for special checks. Florencio and Herley [23], and Egelman et al. [14] improved this approach by adding the size of the alphabet into the calculation and called the resulting value $\log_2((\text{alpha.size})^{\text{pass.len}})$ the bit length of a password.

However, previous ad hoc metrics (e.g., password entropy and bit length) have recently been shown far from accurate by Weir et al. [24], and they suggested that the approach based on simulating password cracking sessions is more promising. They also developed a novel method that first automatically derives word-mangling rules from password datasets, and then instantiates the derived grammars by using string segments from external input dictionaries to generate guesses in decreasing probability order [11]. This PCFG-based cracking approach is able to crack 28% to 129% more passwords than JTR when given the same number of guesses. It has been considered as one of the state-of-the-art password cracking techniques and used in a number of recent works [6], [7].

Differing from the PCFG-based approach, Narayanan and Shmatikov [25] suggested a template-based model in which the Markov-Chain theory is used for assigning probabilities to letter segments, and it substantially reduces the password search space. This approach was tested in an experiment against 142 real-life user passwords and was able to break 67.6% of them. In 2014, by using various normalization and smoothing techniques from the natural language processing (NLP) domain, Ma et al. [7] systematically evaluated the Markov-based model and found that, in some cases, it performs significantly better than the PCFG-based model at large guesses (e.g., $2^{30}$) when parameterized appropriately. Based on [7], [11], some useful automated password analysis tools have been developed in [6], [26]. In this work, we will perform extensive experiments by using both attacking models (combined with the identified characteristics of Chinese passwords) to evaluate the strength of Chinese passwords.

III. A survey on password management behaviors of Chinese users

To reveal Chinese user behaviors and sentiment in the process of password management (e.g., creation and memorization), we carried out an anonymized user survey by using Sojump (www.sojump.com), the most popular Chinese online survey platform, during June 07 to August 11 in 2015. A copy of our questions can be found at http://www.sojump.com/jq/6443561.aspx, and the third part relates to this work. We got approval from our center’s IRB for this survey. The participants are surveyed anonymously and the results are all aggregated such that no user identifiable information can be inferred.

A. Survey design

Though our survey is an anonymized one and conducted on a well-known third-party platform, we fear that users may have privacy concerns and refuse to answer or provide untruthful answers, because passwords are highly sensitive. To minimize such potential adverse effects and ensure ecological validity, we optimize the survey questions before the release of the survey by employing 15 students in other teams of our lab as pilot testers. We get feedbacks from them and revise the questions accordingly. This process is performed in an iterative manner. As a result, a few questions (e.g., “Will you reuse an existing email password for this new email account?” and “How many PIN codes do you maintain?”) that are inclined to be offensive are abandoned. In addition, as with [27], we include an “I prefer not to answer” option for questions that might make participants un-comfortable, in order to avoid user drop out.

Our survey is expected to take about 22 to 35 minutes to complete. Each member in our team is requested to invite 10 to 20 reliable friends, relatives or colleagues to fill out
the questionnaire. If the participants are willing to further disseminate the survey, they are asked to provide us with the number of invitations they send. In all, a total of 983 invitations are sent and we acquire 442 effective responses.

We initiate our survey by asking participants to create a password for an email account that will be frequently used. The first seven questions explore the transformation rules in password reuse (e.g., “how similar the password created for this new account is similar to your existing passwords?”), “what transformation rules will be used?” and “what’s their popularity?”), and detailed results are referred to [9]. The next thirteen questions are devoted to explore user password behaviors regarding semantics composition, overall-reuse, storage, evolution and sentiment, which are the focuses of this work.

B. Survey results

Our first question aims to figure out what’s the semantics of the letters in a user’s password created for this new 163 email account. As shown in Fig. 1, the most popular choice is “Chinese Pinyin names” (about 40%)¹, followed by “Chinese Pinyin words” (33.71%). These two figures well accord with what we have obtained from real-life datasets, i.e. 50.13% and 73% ≈30.02% (see Table V), respectively. Interestingly, many Chinese users also tend to include English words (24.43%) or their English names (22.17%) into their passwords, while these two figures obtained from the large-scale empirical data (see Table V) are 10.74% (≈22.12%) and 23.86% (≈5.35%), respectively. This suggests that users in our survey are one times more likely to use English words than ordinary users do. A plausible reason is that our users are more educated (i.e., 92% have at least a bachelor’s degree, see Sec. III-C), and they are more familiar with English than common user population.

¹Chinese Pinyin is made of 26 Latin characters in the Chinese phonetic alphabet, and they are used to latinate the Chinese hieroglyphic characters. For instance, Chinese Pinyin “woaini” means “iloveyou.”

It has been shown in [7], [17] that Chinese users like to build passwords with digits. Are there semantics in these digits? Fig. 2 shows that 44.8% of users tend to use dates, which well consists with what we have obtained from real-life datasets, i.e. 40.49% (see Table V). In addition, 42.3% use phone numbers (either mobile or fixed-line), 16.74% use homophone (e.g., 5201314) and 3.62% use the time of site registration. What’s most interesting is that, 14.03% Chinese users include their personal digit numbers (PINs) of various financial cards into their passwords, and this figure is unbelievably consistent with Bonneau’s PIN survey on English users (i.e., 15%). This, for the first time, provides more than anecdotal evidence of to what extent Chinese users reuse PINs of financial cards in passwords of web accounts. There are also 27.31% of users choose random digit sequences. Further considering that there are about 1.5 times more Chinese users include digits in passwords than English users do (see Table IV), this partially explains why Chinese web passwords are more difficult to crack at large guess numbers as compared to English passwords (see the “reversal principle” in Sec. V). Besides dates, the other six semantics are inherently difficult to be captured from empirical datasets, highlighting the superior aspect of a user study survey over an empirical analysis.

As our participants are more diversified than that of [28] and our participant number is also two times higher, the observation that Chinese users reuse passwords less is unlikely due to survey bias. A plausible reason may be that, a large number of breached Chinese sites (e.g., over 100 popular sites had leaked passwords during 2011 to 2014 [29]) offer security advices and request users not to reuse passwords. This climax came when China Central Television screened some “best password tips” at peak time in Dec. 2014 [30]. Such advices may increase users’ security consciousness and nudge users towards less direct password reuse, yet the rate of indirect password reuse is still staggering: 57.69% and 74.05% of Chinese users (see Fig. 4) keep no more than 3 and 5 different types of passwords, respectively. In average, each Chinese user keeps 9.74 unique passwords and 3.15 different types of passwords. This is roughly comparable to English users (e.g. 5 unique passwords in [19] and 6.5 unique passwords in [23]).

Fig. 5 illustrates whether there are relationships between users’ different types of passwords. As expected, the majority

---

Fig. 1. If you use letters in your password, what are they related to (multiple-choice)? Most users prefer meaningful strings like Pinyin names and words.

Fig. 2. If you use digits in your password, what are they related to (multiple-choice)? Dates and phone numbers are most prevalent, PINs also show up.

Fig. 3. How many different passwords do you maintain? Note that password1 and password2 shall be seen as different ones. Over 80% of Chinese users are with 10 or fewer unique passwords.

Fig. 4. How many different types of passwords do you maintain? Note that similar passwords fall into the same type. The majority have 3 or fewer types.
of users (60.63%) acknowledge some relationships. In the remains, 21.72% show very good practice, while 17.65% are at high risks: all their passwords are created from a single basic password. We further examine whether Chinese users rationally use different types of passwords for different types of online accounts as recommended [30]. Fig. 6 shows that, 28.05% conform to the good practice, about half of users report they only sometimes conform to the good practice, while 15.84% acknowledge a “definitely no”. This is comparable to Haque et al.’s survey on 80 English students [31]: 19.1% of users reuse (or simple modify) an existing password from a low-value account for a higher-value account.

Fig. 6. Do you use different types of passwords for different types of web services? Less than one third say definitely yes.

The next two questions relate to user behaviors in password storage. Regarding how users saving their passwords, Fig. 7 shows that the most popular way is saving passwords in mind (54.3%), successively followed by writing on paper (25.57%), recording electronically (21.72%), saving in browser (12.9%) and using password manager (7.24%). Komanduri et al. [32] found that in their email survey scenario (1000 English participants), about 19% store passwords on paper and 25% store electronically. In comparison, the difference in writing on paper between their and our user group is significant (two-proportion Z-test, \( p = 0.0048 < 0.05 \)), while the difference in recording electronically is not significant (\( p \)-value= 0.18024 > 0.05).

Fig. 7. How do you save your passwords (multiple-choice)? Over half of users memorize all passwords in mind, while 30% memorize some in mind.

While password managers are deemed promising tools for alleviating the password management burden on users by security researchers [3], it remains a question of to what extent they have been accepted by the common users in reality. Fig. 8 shows that, somewhat surprisingly, over 50% of Chinese users even have not heard about password managers. While 22.17% know password managers, yet they will not use them because of security concerns. Only 13.8% have used them. What’s most staggering is that, only 2.49% of Chinese users have used password managers and also trust their security. Further considering that our participants are more educated and younger than common Chinese users (see Sec. III-C), the acceptance of password managers in China will be even much lower (probably less than 1%). In a survey on English users, Stobert and Biddle were also “surprised to find that none of our participants used a dedicated password manager” [19]. All this suggests that there is still a long way to go before password managers can be widely accepted, especially when considering users’ serious security concerns.

The next three questions relate to how users’ passwords evolve as time goes on. As shown in Fig. 9, 43.22% of users acknowledge little or no difference, while 54.53% show there are tangible differences. The latter well accords with the statistics shown in Fig. 10 that 58.6% of users acknowledge a tangible strength increase in their current passwords over previous ones. Disturbingly, 28.96% admit no strength increase, while the password cracking techniques (e.g., [7], [33]) are evolving rapidly as time goes by. Fig. 11 illustrates how Chinese users cope with diversified password requirements from various sites: 47.74% of users cope in a temporary manner, while 41.63% have developed their own coping method as time goes by.

Fig. 9. Is there any difference between the passwords you have used before and the ones currently in use? 54.53% show there are tangible differences.

The large fraction of users coping with password requirements in a passive manner partially explains why 69.01% of users (see Fig. 12) feel upset/very upset in their management of passwords. Interestingly, there are 15.38% of users show a casual emotion and in the meantime, there are also 11.31% show a positive emotion. This is mainly because there are some users who do not care their online security at all [34] and some users are competent in managing security affairs [19].

Fig. 10. Which are more secure? Your previous passwords or your present passwords? 58.6% acknowledge a tangible strength increase in present ones.

The final question aims to explore whether Chinese users are rational when confronted with stringent password policies from a site. Fig. 13 illustrates that 41.86% of users are rather rational: the necessity of strict policies depends on the importance of the value (service) to be protected. This implies that, from the user prospective, it is unwise for unimportant sites to impose
strict password policies. Unfortunately, our recent large-scale investigation (see [15]) into the leading sites shows quite the opposite: important services (e.g., email and e-commerce) often impose a loose policy, while less important services (e.g., IT corporation and web portal) often enforce a strict policy.

C. Demographics of participants

We also obtained some basic demographic information (see Figs. 14~16) about our participants: two-thirds are male. 80.55% are between 18~34 years old, and 15.67% are older than 35; 80.55% are pursuing or with at least a bachelor’s degree, and 43.44% are pursuing or with at least a master’s degree. This is as expected, for the majority of our team members are young teachers and PhD candidates, and thus their personal relationships are mainly young and educated people. Comparatively, our participants are larger in size and more diversified than that of [19], [28], [31]. For instance, in [28] 93% of its 220 participants are 18~34 years old and 92% have at least a bachelor’s degree. Nevertheless, our participants are younger and more educated than the common population, and they may be more security-conscious and technically savvy. Hence, it is likely that we are underestimating many problems such as password reuse, slow evolution and negative emotion.

Summary. To our knowledge, this survey is the first one that targets Chinese users, the world’s largest Internet population. From the 442 responses, we have revealed a wide variety of useful information (e.g., 14.03% reuse PINs in passwords, over 80% maintain 10 or fewer unique passwords, 54.3% memorize all their passwords, password managers are seldom used and as high as 70% show an unpleasant feeling in managing passwords), providing a deeper understanding of Chinese user password behaviors. Such information is inherently difficult to be obtained from an empirical analysis. Though our participants are superior to earlier related studies (on English users [19], [28], [31]) in terms of size and diversity, yet our survey is subject to the inherent limitation of any password-related survey: users may provide false data and the ecological validity is hard to be assured. Still, in many aspects (e.g., the rate of usages of Pinyin names, Pinyin words, and dates) our results comply with real-life datasets (see Table V) and with English users (e.g., the rates of PIN reuse and recording passwords electronically), implying the validity of our results.

IV. CHARACTERISTICS OF CHINESE PASSWORDS

In this section, we perform an empirical analysis to explore a number of Chinese password characteristics that have received little attention in the literature.

A. Dataset descriptions and ethics consideration

Our empirical analysis employs six Chinese web password datasets and three English password datasets. In total, these nine datasets consist of 130 million passwords, one of the largest corpuses ever studied. As summarized in Table I, these nine datasets are different in terms of service, language/culture, size and user localization. They were hacked by external attackers or disclosed by anonymous insiders, and were subsequently made public on the Internet. More detailed information about these datasets can be found in Appendix A.

We realize that though publicly available, these datasets are private data. Therefore, we only report the aggregated statistical information, and treat each individual account as confidential such that using it in our research will not increase risk to the corresponding victim, i.e., no personally identifiable information can be learned. Furthermore, these datasets may be exploited by attackers as cracking dictionaries, while our use of them is both beneficial for the academic community to understand password choices of Chinese netizens and for security administrators to secure user accounts. Also note that, since the datasets we employed are all publicly available, all the results in this work are reproducible.

B. Data cleansing

Before exploring the password characteristics, we perform the process of data cleansing. We get rid of email addresses and user names from the original data. As with [7], we remove strings that include characters beyond the 95 printable ASCII symbols. We further remove strings whose length is over 30, because after having manually scrutinized the original datasets, we find that these long strings do not seem to be generated by human beings, but more likely by password managers or simply junk information. Moreover, such unusually long passwords are often beyond the scope of attackers who specially care about cost-effectiveness [6], [18]. In all, the fraction of excluded passwords is negligible, yet this cleansing step unifies the input of cracking algorithms and simplifies the later data processing.

Of particular interest is our observation that, there is a non-negligible overlap between the Tianya dataset and 7k7k dataset. We were first puzzled by the fact that the password “111222tianya” originally lay in the top-10 most popular list of both datasets. We manually scrutinize the original datasets (i.e., before removing the email addresses and user names) and are surprised to find that there are around 3.91 million (actually 3.91*2 million due to a split representation of 7k7k accounts, as we will discuss later) joint accounts in both datasets. We realize that someone probably have copied these joint accounts from one dataset to the other.

Now, a natural question arises: From which dataset have these joint accounts been copied? It is highly likely that these joint accounts were copied from Tianya to 7k7k mainly for two reasons. Firstly, it is unreasonable for 0.34% users in 7k7k to insert the string “tianya” into their 7k7k passwords, while users from tianya.cn are natural to include the site name “tianya” into their passwords for convenience. The following second reason is quite subtle yet convincing. In the original Tianya dataset, we find that the joint accounts are of the form {user name, email address, password}, while in the original 7k7k dataset such a joint account is divided into two parts: {user name, password} and {email address, password}. The password “111222tianya” occurs 64822 times in 7k7k and 48871 times in Tianya, and one gets that $64822 < 48871$. Therefore, it is more plausible for someone to copy some (i.e., 64822/2 of a total of 48871) accounts using “111222tianya” as the password from Tianya to 7k7k, rather than to copy all the accounts (i.e., 64822/2) using “111222tianya” as the password from 7k7k to Tianya and further reproduces $16460(= 48871 – 64822/2)$ such accounts.
After removing 7.82 million joint accounts from 7k7k, we found that all of the passwords in the remaining 7k7k dataset occur even times (e.g., 2, 4 and 6). This is expected, for we observe that in 7k7k half of the accounts are of the form {user name, password}, while the rest are of the form {email address, password}, and it is likely that both forms are directly derived from the form {user name, email address, password}. For instance, both {wanglei, wanglei123} and {wanglei@gmail.com, wanglei123} are actually derived from the single account {wanglei, wanglei@gmail.com, wanglei123}. Consequently, we further divide 7k7k into two equal parts and discard one part. The detailed information on data cleansing is summarized in Table 1.

In 2014, Li et al. [17] has also exploited the datasets Tianya and 7k7k. However, contrary to what we have done above, they think that the 3.91M joint accounts are copied from 7k7k to Tianya. Their mere reason is that, when dividing these two datasets into the reused passwords group (i.e., the joint accounts) and the not-reused passwords group, they find that “the proportions of various compositions are similar between the reused passwords and the 7k7k’s not-reused passwords, but different from Tianya’s not-reused passwords”. However, they have never explained what these “various compositions” are. Their explanation also cannot answer the critical question: why are there so many 7k7k users using “vegetable” as their passwords? Hence, it would be better that they had removed 3.91*2 million joint accounts from 7k7k but not 3.91 million ones from Tianya. In addition, they did not find that all the passwords in 7k7k occur even times, which is extremely abnormal. Such contaminated data would highly lead to inaccurate results and unreliable comparisons. For example, Li et al. reported that there are 9,477,069 (30.67%) passwords in Tianya with consecutive exactly six digits, yet the actual value is 2.5 times larger: 23,358,248 (75.59%).

### C. Password characteristics

It is widely hypothesized that user-generated passwords are greatly influenced by their native languages, yet so far little quantitative measurement has been given. To fill this gap, here we first illustrate the character distributions of the nine password datasets, and then measure the closeness of passwords with their native languages in terms of inversion number of the character distributions (in descending order).

As expected, passwords from different language groups have significantly varied letter distributions (see Fig. 17). What’s unexpected is that, even though generated and used in vastly diversified web services, passwords from the same language group have quite similar letter distributions. This suggests that, when given a password dataset, one can largely determine what’s the native language of its users by investigating its letter distribution. Arranged in descending order, the letter distribution of all Chinese passwords is aineohglwyszqcdjmbtkpv, while this distribution for all English passwords is aeionrlstcmdbhkgp jvwfxzq. While some letters (e.g., ‘a’, ‘e’ and ‘i’) occur frequently in both groups, some letters (e.g., ‘q’ and ‘c’) only occur frequently in one group. Such information can be exploited by attackers to reduce the search space and optimize their cracking strategies. Note that, here all the percentages are handled case-insensitively.

While users’ passwords are greatly affected by their native languages, the letter frequencies in general language usages may be somewhat different from the frequencies of letters in passwords. To what extent do they differ? According to Huang et al.’s work [35], the letter distribution of Chinese language (i.e., written Chinese texts like literary work, newspapers and academic papers), when converted into Chinese Pinyin, is inauhegyoazdjmxwgbctlprkv. This shows that some letters (e.g., ‘l’ and ‘w’), which are popular in Chinese passwords, appear much less frequently in written Chinese texts. A plausible reason may be that ‘l’ and ‘w’ is the first letter of the family name li and wang (which are the top-2 family names in China), respectively, while Chinese users, as we will show, love to use names to build their passwords.

Similar observation also holds for English passwords. The letter distribution of English language (i.e., etaoinsrhdl cumwfgypbvkjxqz) is obtained from www.cryptograms.org/letter-frequencies.php. For instance, the letter ‘t’ is common in English texts, but not so in English passwords. A plausible reason may be that ‘t’ is used in popular words such as the, it, this, that, at, to, while such words are not common in passwords.

### TABLE I. DATA CLEANSING OF THE NINE PASSWORD DATASETS

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Web service</th>
<th>Location</th>
<th>Language</th>
<th>Original</th>
<th>Miscellany</th>
<th>Length&gt;30</th>
<th>All removed</th>
<th>After cleansing</th>
<th>Unique passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tianya</td>
<td>Social forum</td>
<td>China</td>
<td>Chinese</td>
<td>31,701,424</td>
<td>800,178</td>
<td>3</td>
<td>2.71%</td>
<td>30,901,241</td>
<td>12,589,457</td>
</tr>
<tr>
<td>7k7k</td>
<td>Gaming</td>
<td>China</td>
<td>Chinese</td>
<td>19,138,452</td>
<td>13,705,087</td>
<td>10,078</td>
<td>71.66%</td>
<td>5,423,282</td>
<td>2,863,573</td>
</tr>
<tr>
<td>Dodonew</td>
<td>Gaming&amp;E-commerce</td>
<td>China</td>
<td>Chinese</td>
<td>16,283,140</td>
<td>10,774</td>
<td>13,475</td>
<td>0.15%</td>
<td>16,258,891</td>
<td>10,135,260</td>
</tr>
<tr>
<td>178</td>
<td>Gaming</td>
<td>China</td>
<td>Chinese</td>
<td>9,072,966</td>
<td>0</td>
<td>1</td>
<td>0.00%</td>
<td>9,072,965</td>
<td>3,462,283</td>
</tr>
<tr>
<td>CSDN</td>
<td>Programmer forum</td>
<td>China</td>
<td>Chinese</td>
<td>6,428,632</td>
<td>355</td>
<td>0</td>
<td>0.01%</td>
<td>6,428,277</td>
<td>4,037,605</td>
</tr>
<tr>
<td>Duowan</td>
<td>Gaming</td>
<td>China</td>
<td>Chinese</td>
<td>5,024,764</td>
<td>42,024</td>
<td>10</td>
<td>0.83%</td>
<td>4,982,730</td>
<td>3,119,060</td>
</tr>
<tr>
<td>Rockyou</td>
<td>Social forum</td>
<td>USA</td>
<td>English</td>
<td>32,603,387</td>
<td>18,377</td>
<td>3140</td>
<td>0.07%</td>
<td>32,581,870</td>
<td>14,326,970</td>
</tr>
<tr>
<td>Yahoo</td>
<td>Portal(e.g., E-commerce)</td>
<td>USA</td>
<td>English</td>
<td>453,491</td>
<td>10,657</td>
<td>0</td>
<td>2.35%</td>
<td>442,834</td>
<td>342,510</td>
</tr>
<tr>
<td>Phpbb</td>
<td>Programmer forum</td>
<td>USA</td>
<td>English</td>
<td>255,421</td>
<td>45</td>
<td>3</td>
<td>0.02%</td>
<td>255,373</td>
<td>184,341</td>
</tr>
</tbody>
</table>

*We remove 13M duplicate accounts from 7k7k, because we identify that they are copied from Tianya as we will detail in Section IV-B.

![Fig. 17. Letter distributions of passwords](image1.png)

![Fig. 18. Length distributions of passwords](image2.png)

![Fig. 19. Frequency distributions of passwords](image3.png)
To further explore the closeness of passwords with their native languages and with the passwords from other datasets, we measure the inversion number of the letter distribution sequence (in descending order) between two password datasets (as well as languages), and the results are summarized in Table II. “Pinyin_fullname” is a dictionary consisting of 2,426,841 unique Chinese full names (e.g., wangle and zhangwei), “Pinyin_word” is a dictionary consisting of 127,878 unique Chinese words (e.g., chang and cheng), and these two dictionaries will be detailed later. Note that the inversion number of sequence A to sequence B is equal to that of B to A. For instance, the inversion number of inanh to anikh is 3, which is equal to that of anikh to inanh.

Table II illustrates that, the inversion number of letter distributions between passwords from the same language group is generally much smaller than that of passwords from different language groups. This value is also distinctly smaller than that of the letter distributions between passwords and their native language (see the bold values in Table II). All these indicate that passwords from different languages are intrinsically different from each other in letter distributions, and that passwords from the same language group are close to their native language yet the distinction is still noticeable (measurable).

Note that, among all Chinese datasets, Duowan has the least inversion number (i.e., 9 in dark gray) with the dataset “All_Chinese_PW’s”. This indicates that Duowan is likely to best represent general Chinese password datasets, and thus it will be selected as the training set for attacking other Chinese datasets (see Sec V). For a similar reason, Rockyou will be selected as the training set for attacking English passwords.

Fig. 18 depicts the length distributions of passwords. Irrespective of the web service, language and culture differences, the most common password lengths of every dataset are between 6 and 10, among which length-6 or 8 takes the lead. Merely passwords of these five lengths can account for more than 75% of every entire dataset, and this value will rise to 90% if we consider passwords with lengths of 5 to 12. As expected, very few users prefer passwords longer than 15 characters. Notably, people seem to prefer even length over odd length. Another interesting observation is that, CSDN exhibits only one peak in its length distribution curve and has much fewer passwords (i.e., only 2.16%) in lengths below 8. This is likely due to the fact that this site has enforced a minimal length-8 policy since an early stage.

Fig. 19 portrays the frequency vs. the rank of passwords from different datasets in a log-log scale. We first sort each dataset according to the password frequency in descending order, and then each individual password will be associated with a frequency \( f_i \) and a rank \( r_i \). Interestingly, the curve for each dataset closely approximates a straight line, and this trend will be more pronounced if we take all the nine curves as a whole. This well corroborates the Zipf theory [36]: \( f_i \) and \( r_i \) follow a relationship of the type \( \log f_i = -s \cdot \log r_i \), where \( C \) and \( s \) are constants. Particularly, \( s \) is the absolute value of slope of the Zipf linear regression line and slightly less than 1.0. The Zipf theory indicates that the popularity of user-generated passwords decreases polynomially with the increase of their rank. This further implies that a few passwords are overly popular (and thus online guessing can be effective), while the least frequent passwords are very sparsely scattered in the password space (and thus the offline guessing attackers need to consider cost-effectiveness and weigh when to stop).

Table III shows the top-10 most frequent passwords from different services. The most frequent password among all datasets is “123456”, with CSDN being the only exception, which is likely due to the CSDN password policy requiring passwords to be at least 8 characters. “111111” follows on the heel. Other popular Chinese passwords include “123123”, “123321”, which are all composed of digits and in simple patterns such as repetition. Love also shows its magic power: “5201314”, which sounds like “I love you forever and ever” in Chinese, appears in the top-10 lists of four Chinese datasets.

In contrast, popular ones in English datasets tend to be meaningful letter strings (e.g., “sunshine” and “letmein”). The eternal theme of love — frankly, “iloveyou” or euphemistically, “princess” — also show up in top-10 lists of English datasets. Our results confirm the folklore that “back at the dawn of the Web, the most popular password was 123456. Today, it is one digit longer but hardly safer: 123456.”

It is worth noting that, Chinese passwords are highly con-
centrated, because only the top-10 most popular ones amount to as high as 6.78%~10.44% of each entire dataset, with DodoneW being the mere exception. However, even DodoneW achieves 3.24%, while the English datasets are all below 2.80%. This implies that Chinese passwords are more prone to online guessing, which will be confirmed in Sec. V.

As we have seen that digits are popular in top-10 passwords of Chinese datasets, are they also popular in the whole datasets? To answer this question, we investigate the frequencies of password patterns that involve digits, and results on the top 3 most frequent ones are shown in the left hand of Table IV. More results (i.e., on the top-10 ones) can be found in Appendix B (see the supplemental file). The first column of the table denotes the pattern of a password as in [11] (i.e., L denotes a lower-case sequence, D for digit sequence, U for upper-case sequence, S for symbol sequence, and the structure pattern of password “Wanglei123” is ULD). We see that an average of more than 50% of Chinese web passwords are only composed of digits, while this value of English datasets is only 11.30%. In contrast, English users prefer the patterns L and LD.

It is somewhat surprising to see that, the sum of the three digit-based patterns (i.e., D, LD and DL) accounts for an average of 81.90% for Chinese datasets. In contrast, English users favor letter-related patterns, and on average, their top-3 patterns (i.e., L, LD and D) also account for slightly over 80%. This indicates that, unlike English users, Chinese users are inclined to employ digits to build their passwords — digits serve the role of letters that play in English passwords, while letters mainly come from Chinese Pinyin words/names. This is probably due in large part to the fact that most Chinese users are unfamiliar with English language (and Roman letters on the keyboard). If this is the case, is there any meaningful (semantic) information underlying these digit sequences?

To gain an insight into the underlying semantic patterns, we construct 22 dictionaries of different semantic categories and investigate their prevalence. More detailed information about these 22 dictionaries are referred to Appendix C. Table V shows the various semantic patterns existing in Chinese and English passwords. We can see that, a large fraction of English users tend to use raw English words as their password building blocks. More specially, 25.88% English users insert a 5-letter or longer (denoted by 5\+-letter) word into their passwords, and this figure accounts for over a third of the total passwords with a 5\+-letter substring. In comparison, few Chinese users choose raw Pinyin words or English words to build passwords, yet they prefer Pinyin names, especially full names.

Particularly, of all the Chinese passwords (22.42%) that contain a 5\+-letter substring, more than half (11.24%) include a 5\+-letter Pinyin full name. This well complies with that of our user survey. There is also a non-negligible proportion (i.e., 4.10%) of English passwords that contain a 5\+-letter full Pinyin name, and a reasonable explanation is that many Chinese users have created accounts in these English sites. For instance, the popular Chinese name “zhangwei” appears in both Rockyou and Yahoo. We also note that English names are also widely used in English passwords, yet full names are less popular than last names and first names. As far as we know, for the first time we have explored name-based semantic patterns in a large-scale empirical password study.

Equally interestingly, we find that, on average, 16.99% of Chinese users insert a six-digit date into their passwords. Further considering Fig. 2, such dates are likely to be users’ birthdays. Besides, about 30.89% of Chinese users employ a 4\+-digit date as their password building blocks, which is 3.59 times higher than that of English users (i.e. 8.61%); there are 13.49% of Chinese users inserting a four-digit year into their passwords, which is about 3.55 times higher than that of English users (3.80%, which is comparable to the results reported in [28]). We note that there might be some overestimates, for there is no way to definitely tell apart whether some digit sequences are dates or not, e.g., 010101 and 520520. These two sequences may be dates, yet they are also likely to be of other semantic meanings (e.g., 520520 sounds like “I love you…”). As discussed later, we have devised reasonable ways to address this issue. In all, dates play a vital role in Chinese user passwords.

Another interesting observation is that, 2.91% of Chinese users just use their 11-digit mobile numbers as passwords, making up 39.59% of all passwords with a 11\+-digit substring. In average, 12.39% of Chinese passwords are longer than 11. Thus, if an attacker can determine (e.g., by shoulder-surfing) that the victim uses a long password, she is likely to succeed with a high chance 23.48%=2.91% by just trying the victim’s 11-digit mobile number. This reveals a practical attacking strategy against long Chinese passwords.

Note that there are some un-avoidable ambiguities when determining whether a text/digit sequence belongs to a specific dictionary, and improper resolution of these ambiguities would lead to an overestimation or underestimation of human choices. Here we take the dictionary “YYMMD” for illustration. For example, both 111111 and 520521 fall into “YYMMD” and are highly popular, yet it is more likely that users choose them simply because they are easily memorable repetition numbers or meaningful strings, and counting them as dates would lead to an overestimation. Yet, they can really be dates (e.g., 1111111 stands for “Jan. 1th, 2011” and 520521 stands for “May 21th, 1952”), and completely excluding them from “YYMMD” would result in an underestimation of dates.

Thus, we assume that user birthdays are randomly distributed, and assign the expectation of frequency of dates (denoted by $E$), instead of zero, to the frequency of these abnormal dates. We manually identify 17 abnormal dates in the dictionary “YYMMD”, each of which is originally with a frequency greater than $10E$ and appears in every top-1000 list of the six Chinese datasets. In this way, the dilemma can be largely resolved. We similarly tackle 21 abnormal items in “MMDD”. As for the other 19 dictionaries in Table V, few abnormal items can be identified, and thus they are processed as usual.

Summary. We have measured nine password datasets (including 6 Chinese ones and 3 English ones) in terms of letter distribution, length distribution, frequency distribution and semantic patterns. To our knowledge, most of these fundamental characteristics have not been examined in the literature. We have identified a number of similarities (e.g., frequency
We divide the nine datasets into two groups according to the languages involved. For the Chinese group of test sets, we randomly select 1M passwords from the Duowan dataset as the training set (denoted by “Duowan_1M”). The reason why we select Duowan as the training set has been discussed in Sec. IV-C. For the English group of test sets, for a similar reason we select 1M passwords from Rockyou as the training set. More rationale underlying our choices of these two as training sets is that, passwords in Duowan and Rockyou exhibit more composition varieties than that of other datasets in the same group. This can be seen from Table III to V. Since we have only used part of Duowan and Rockyou, their remaining passwords as well as the other seven datasets are used as the test sets. The attacking results on the Chinese group and English group are depicted in Fig. 20(a) and Fig. 20(b), respectively.

One can see that, when the guess number (i.e., search space size) allowed is below about 3,000, Chinese passwords are generally much weaker than English passwords from the same service (i.e., Tianya vs. Rockyou, Dodonew vs. Yahoo, and CSDN vs. phpbw). For example, at 100 guesses, the success rate against Tianya, Dodonew and CSDN is 10.2%, 4.3% and 9.7%, respectively, while their English counterparts are 4.6%, 1.9% and 3.7%, respectively. However, when the search space size is above 10,000, Chinese web passwords are generally much stronger than their English counterparts. For example, at 10 million guesses, the success rate against Tianya, Dodonew and CSDN is 37.5%, 28.8% and 29.9%, respectively, while their English counterparts are 49.7%, 39.0% and 41.4%, respectively. The strength gap between these two groups of datasets will be even wider when the guess number further increases. This reveals that a reversal principle has occurred: Chinese passwords are more vulnerable to online guessing attacks (i.e., when the guess number allowed is small), while English passwords are more prone to offline guessing attacks in which the attacker is not subject to the restriction of the guess number. This “reversal principle” well reconciles two drastically conflicting claims (see Section I-A) made about the security strength of Chinese web passwords.

We observe that, the original PCFG-based algorithm [7], [11] inherently gives extremely low probabilities to password guesses (e.g., “1q2w3e4r” and “1a2b3c4d”) that
are of a monotonically long structure (e.g., $D_1 L_1 D_1 L_1 D_1 L_1 D_1 L_1$, or $(D_1 L_1)_4$ for short). For example, $P(^{1q2w3e4z^*}) = P((D_1 L_1)_4) \cdot P(D_1 \rightarrow \omega) \cdot P(L_1 \rightarrow \omega) \cdot P(D_1 \rightarrow e) \cdot P(D_1 \rightarrow e) \cdot P(D_1 \rightarrow 4) \cdot P(L_1 \rightarrow r)$ can hardly be larger than $10^{-9}$, for it is a multiplication of nine probabilities. As a result, some guesses (e.g., “1q2w3e4z”) will never appear in the top $10^7$ guesses generated by the original PCFG-based algorithm, even though they are popular (e.g., “1q2w3e4z” appears in the top-200 list of every dataset). The essential reason is that the PCFG-based algorithm simply assumes that each segment in a structure is independent. Yet, in many situations this is not true. For instance, the four $D_1$ segments and $L_1$ segments in the structure $(D_1 L_1)_4$ of password “1q2w3e4z” are evidently interrelated with each other.

To address this problem, we specially tackle a few structures that are long but simple alternations of short segments by treating them as short structures, e.g., $(D_1 L_1)_4$ and $(D_1 L_1)_3$ are converted to $D_1 L_1$ and $D_1 L_1$, respectively. In this way, the probability of “1q2w3e4z” now is computed as $P(^{1q2w3e4z^*}) = P((D_1 L_1)_4) \cdot P((D_1 L_1)_4 \rightarrow D_1 L_1) \cdot P(D_1 \rightarrow 1234) \cdot P(L_1 \rightarrow q\omega e\omega)$. Our approach is language-irrelevant and constitutes a general amendment to the state-of-the-art PCFG-based algorithm in [7].

Further exploit the characteristics of Chinese passwords, we insert the “Pinyin name any” dictionary and the six-digit date dictionary (see Sec. IV-C) into the original PCFG L-segment dictionary and D-segment dictionary, respectively. Details about this insertion process, and our improved algorithm for the generation of password guesses, are shown in Appendix E. The resulting changes to the original PCFG grammars learned from the training sets are shown in Table VI.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Base structures</th>
<th>L segments</th>
<th>D segments</th>
<th>S segments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duowan_1M</td>
<td>8905+0</td>
<td>155934+24416</td>
<td>465157+20341</td>
<td>865+0</td>
</tr>
<tr>
<td>Duowan_All</td>
<td>20961+0</td>
<td>5590174+98654</td>
<td>1824404+9744</td>
<td>2417+0</td>
</tr>
</tbody>
</table>

Fig. 20(c) shows that, when the guess number is small (e.g., $10^3$), our improved attack exhibits little improvement; while the guess number grows, the improvement increases. For example, at $10^7$ guesses, there is $0.09\% \sim 0.85\%$ improvement in success rate; at $1M$ guesses, this figure is $1.32 \sim 4.32\%$; at $10M$ guesses, this figure rises to $1.70\% \sim 4.29\%$. This indicates that, the very popular usages of Pinyin names and birthdays facilitate an attacker to reduce the search space, and this vulnerability is especially serious when large guesses are allowed.

In 2014, Li et al. [17] reported that using 2 million Dodowen passwords as the training set and at 10 billion guesses, their best cracking record is about 17.30% (see Fig. 5 of [17]). However, our improved attack, which uses only 1 million passwords as the training set and at merely 10 million guesses, is able to achieve success rate from 29.41% to 39.47%. This means that our improved attack can crack 70% to 128% more passwords than Li et al.’s best record (i.e., 17.3%, see Fig. 5 of [17]). Alarming high success rates highlight the urgency of developing effective countermeasures (e.g., more practical password creation policies) to alleviate the situation.

In our improved PCFG-based attacks, external name segments are added into the PCFG L-segment dictionary during training, and we get gladsome increases in success rates (see Figs. 20(c)). However, such improvements are still not so prominent as compared to the prevalence of names in Chinese passwords. To explicate this paradox, we scrutinize the internal process of PCFG-based guess generation and manage to identify its crux. Here we take the improved PCFG-based attack against Tianya (using Duowan as the training set) as an example. During training, we have added 98K name segments (see Table VI) into the L-segment dictionary. As shown in Fig. 22(a), these 98K name segments only cover 2.88% of the total L segments of the Tianya test set. However, the original L segments trained from Duowan can cover 13.75% of the name segments and 60.59% of the non-name L segments in the Tianya test set. This suggests that Duowan is able to well cover the name segments in the test set Tianya, and thus the addition of some extra names would be of limited yields. This observation also holds for the other eight test sets, and the detailed results are presented in Appendix F. Note that this does not contradict our conjecture that Pinyin names pose a serious vulnerability, and actually, it does suggest that when the training set is selected properly, the name segments in passwords can be well represented. Still, when there is no proper training set available, our improved attack would show its advantages (see Fig. 22(b)). Moreover, although our improved PCFG-based algorithm might not be optimal, its cracking results represent a new benchmark that any future algorithm should aim to decisively clear.

---

**Fig. 22.** Coverage and security impact of Pinyin-name-segments in the test set Tianya (using Duowan as the training set and Pinyin name as an extra input dictionary when performing our improved PCFG-based attack).
B. Markov-Chain-based attacks

In our Markov-Chain-based password cracking experiments, as recommended in [7], we consider two smoothing techniques (i.e., Laplace Smoothing and Good-Turing Smoothing) to deal with the data sparsity problem, two normalization techniques (i.e., distribution-based and end-symbol-based) to deal with the unbalanced length distribution problem of passwords. This brings about four attacking scenarios as listed in Table VII. In each scenario we consider three types of markov order (i.e., order-5, 4 and 3) to investigate which order performs best. It is reported in [7] that another scenario (i.e., backoff with end-symbol normalization) performs “slightly better” than the aforementioned four scenarios, yet it is approximately 11 times slower, both for guess generation and for probability estimation” [7]. We also investigate this scenario and observe similar results in our experiments. Therefore, attackers, who particularly care about the cost-effectiveness, are highly unlikely to exploit this scenario. Due to space constraints, the detailed password guess generation procedure for Scenario #1 is referred to Algorithm I in the supplemental material, and the generation procedures for the other scenarios are quite similar.

<table>
<thead>
<tr>
<th>Attack order</th>
<th>Smoothing</th>
<th>Normalization</th>
<th>Markov order</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>Laplace</td>
<td>End-symbol</td>
<td>3/4/5</td>
</tr>
<tr>
<td>#2</td>
<td>Laplace</td>
<td>Distribution</td>
<td>3/4/5</td>
</tr>
<tr>
<td>#3</td>
<td>Good-Turing</td>
<td>End-symbol</td>
<td>3/4/5</td>
</tr>
<tr>
<td>#4</td>
<td>Good-Turing</td>
<td>Distribution</td>
<td>3/4/5</td>
</tr>
<tr>
<td>#5</td>
<td>Backoff</td>
<td>End-symbol</td>
<td>Backoff</td>
</tr>
</tbody>
</table>

As with PCFG-based attacks, in our implementation we use a max-hep to store the interim results to maintain efficiency. To produce $k = 10^7$ guesses, we employ the strategy of first setting a lower bound (i.e., $10^{-9}$) for the probability of guesses generated, then sorting all the guesses and finally selecting the top $k$ ones. In this way, we are able to reduce the time overheads by 170% at the cost of about 110% increase in storage overheads, as compared to the strategy of producing exactly $k$ guesses. In Laplace Smoothing, it is required to add $\delta$ to the count of each substring and we set $\delta = 0.01$ as suggested by Ma et al. [7]. The cracking results for Scenario #1 are included in Fig. 21. Due to space constraints, the experiments for attacking scenarios #2~#5 are illustrated in the Appendix G. A subtlety to be noted when implementing the Good-Turing (GT) smoothing technique is referred to Appendix H.

Our experiments (see Fig. 21 and the Appendix G) show that for both Chinese and English test sets: (1) At large guesses (i.e., larger than $2\times10^9$), order-4 markov-chain evidently performs better than the other two orders, while at small guesses (i.e., lower than $10^5$) the larger the order, the better the performance will be; (2) There is not much difference in performance between Laplace and GT Smoothing at small guesses, while the advantage of Laplace Smoothing gets greater as the guess number increases; (3) End-symbol-based normalization always performs better than the distribution-based approach, while at small guesses its advantages will be more obvious. This suggests that at large guesses, the attacks preferring order-4, Laplace Smoothing and end-symbol normalization (see Figs. 21(b) and 21(e)) perform best; At small guesses, the attacks preferring order-5, Laplace Smoothing and end-symbol normalization (see Figs. 21(a) and 21(d)) perform best.

Note that, the reversal principle found in our PCFG-based attacks (see Sec. V-A) also applies in all the Markov-based experiments. For example, in order-4 Markov-based attacks (see Figs. 21(b) and 21(e)), we can see that, when the guess number is below 7000, Chinese passwords are generally much weaker than their English counterparts. For example, at $10^5$ guesses, the success rate against Tianya, Dodonew and CSDN is 11.8%, 6.3% and 11.6%, respectively, while their English counterparts (i.e., Rockyou, Yahoo and Phpb) are merely 8.1%, 4.3% and 7.1%, respectively. However, when the guess number allowed is over $10^4$, Chinese passwords are generally stronger than their English counterparts. For example, at 1M guesses, the success rate against Tianya, Dodonew and CSDN is 38.7%, 21.2% and 25.9%, respectively, while their English counterparts are 39.2%, 26.1% and 33.3%, respectively.

It is interesting to see that CSDN enforces a minimum length-8 policy (as shown in Fig. 18 and [36], 97.83% passwords in CSDN are of length 8+), while Dodonew enforces no apparent rule (i.e., neither minimum length nor character
work provides a deeper understanding of Chinese passwords and languages, frequency distribution and various semantic patterns) that characterize user-chosen passwords. In our knowledge, this survey is the first one that targets Chinese users and perform a large-scale empirical analysis of 100 million real-world Chinese web passwords. To our knowledge, this survey is the first one that targets Chinese users. We get 442 effective responses and reveal a number of users’ basic coping strategies for managing passwords. In our empirical analysis, we, for the first time, explore several fundamental properties (e.g., the distance between passwords and languages, frequency distribution and various semantic patterns) that characterize user-chosen passwords.

Particularly, we evaluate password strength by using two state-of-the-art cracking algorithms and our improved PCFG-based algorithm, and uncover a “reversal principle”: Chinese passwords are more susceptible to online guessing attacks, while English ones are more vulnerable to offline guessing attacks. This, for the first time, well reconciles two conflicting claims [17], [18]. Considering the systematicness and comprehensiveness of our exploration, it is expected that this work provides a deeper understanding of Chinese passwords and reveals substantial “ground-truth” for better design and evaluation of future password policies, strength meters, etc.

VI. Conclusion

In this paper, we carry out a user survey on password behaviors of Chinese users and perform a large-scale empirical analysis of 100 million real-world Chinese web passwords. To our knowledge, this survey is the first one that targets Chinese users. We get 442 effective responses and reveal a number of users’ basic coping strategies for managing passwords. In our empirical analysis, we, for the first time, explore several fundamental properties (e.g., the distance between passwords and languages, frequency distribution and various semantic patterns) that characterize user-chosen passwords.

Particularly, we evaluate password strength by using two state-of-the-art cracking algorithms and our improved PCFG-based algorithm, and uncover a “reversal principle”: Chinese passwords are more susceptible to online guessing attacks, while English ones are more vulnerable to offline guessing attacks. This, for the first time, well reconciles two conflicting claims [17], [18]. Considering the systematicness and comprehensiveness of our exploration, it is expected that this work provides a deeper understanding of Chinese passwords and reveals substantial “ground-truth” for better design and evaluation of future password policies, strength meters, etc.

REFERENCES


APPENDIX A
DETAIL INFORMATION ABOUT THE NINE REAL-WORLD PASSWORD DATASETS

In this work, we have employed nine real-world password datasets. The first three datasets are all from US. The Rockyou dataset [1] includes over 32M passwords and was hacked from the social application site rockyou.com in Dec. 2009 by an SQL injection attack. The Phpbp dataset contains around 255K passwords leaked from phpb.com, a forum on the development of PHP scripting language, in Jan. 2009. The Yahoo dataset [2] consists of about 442K passwords leaked by the hacker group named D33Ds in July 2012.

The following six datasets were all leaked from Chinese sites in Dec. 2011 [3]. The 6.42M CSDN passwords were hacked from csdn.net, a popular community for software developers in China. The 31.76M Tianya passwords were leaked from tianya.cn, an influential Chinese BBS forum. The remaining four datasets are all from popular Chinese gaming websites, of which Dodonew is a site with e-commerce services. As expected, the Dodonew passwords is the strongest one among all datasets (see Section 5 of the main text). It is revealed that users often rationally choose robust passwords for accounts perceived to be of an important value [4], while knowingly choose weak passwords for unimportant accounts [5]. Since accounts of the same service generally would have the same level of value for a user, we divide datasets into three pairs according to their types of service (i.e., Tianya vs. Rockyou, Dodonew vs. Yahoo, and CSDN vs. Phpbp) when performing strength comparison in Section 5 of the main text.

APPENDIX B
TOP-10 MOST POPULAR PASSWORD PATTERNS WITH DIGITS

We have seen that digits are popular in top 10 passwords of Chinese datasets, whether are they popular in the whole datasets? To answer this question, we investigate the frequencies of password patterns that involve digits, and the results (on top 10 most frequent patterns) can be found in Table B.1. The first row of the table denotes the pattern of a password as in [6] (L denotes a lower-case sequence, D for digit sequence, U for upper-case sequence, and S for symbol sequence). We see that an average of more than 50% of Chinese web passwords are only composed of digits, while this value of English datasets is only 15.77%. In contrast, English users prefer the pattern LD. Note that, all the percentages hereafter in this work are taken by dividing the corresponding total accounts (e.g., the percentage at the upper-left corner of Table B.1 is computed as \frac{\text{# of passwords with pattern } D}{\text{# of total passwords in Tianya}} = \frac{19706174}{309901241} = 63.77\%).

APPENDIX C
DETAIL INFORMATION ABOUT THE TWENTY-TWO SEMANTIC DICTIONARIES

To gain an insight into the underlying semantic patterns, we construct 22 dictionaries of different semantic categories and investigate their prevalence (see Table V of the main text). “English_word_lower” is from http://www.mieliestronk.com/wordlist.html and it contains about 58,000 popular lower-case English words. “English_lastname” is a dictionary consisting of 18,839 last (family) names with over 0.001% frequency in the US population during the 1990 census, according to US Census Bureau [7]. “English_firstname” contains 5,494 most common first names (1,219 male and 4,275 female names) in US [7]. “Englishfullname” is a cartesian product of “EnglishFirstname” and “EnglishLastName”, consisting of about 1.04 million most common English full names.

To get a Chinese full name dictionary, we employ the 20 million hotel reservations dataset [8] leaked in Dec. 2013. The Chinese family name dictionary includes 504 family names which are officially recognized in China. Since the first names of Chinese users are widely distributed and can be almost any combinations of Chinese words, we do not consider them in this work. As the names are originally in Chinese, we transfer them into Pinyin without tones by using a Python procedure from https://pypinyin.readthedocs.org/en/latest/ and remove the duplicates. We call these two dictionaries “Pinyinfullname” and “Pinyinfamilyname”, respectively.

“Pinyin_word_lower” is a Chinese word dictionary known as “SogouLabDic dic”, and “Pinyin_place” is a Chinese place dictionary. Both of them are from [9] and also originally in Chinese, and we translate them into Pinyin in the same way as we tackle the name dictionaries. “Mobile_number” consists of all potential Chinese mobile numbers, which are 11-digit strings with the first seven digits conforming to predefined values and the last four digits being random. Since it is almost impossible to build such a dictionary on ourselves, we instead write a Python script and automatically test each 11-digit string against the mobile-number search engine http://ku.13131313131.com/.

As for the birthday dictionaries, we use patterns to match digit strings that might be birthdays. For example, YYYY-MM-DD stands for a birthday pattern that the first four digits indicate years (from 1900 to 2014), the middle two represent months (from 01 to 12) and the last two denote dates (from 01 to 31). “PW with a l±-digit substring” is a subset of the corresponding dataset and consists of all passwords that include a letter substring no shorter than l, and similarly for “PW with a l±-digit substring”.

APPENDIX D
THE NECESSITY OF PAIRING PASSWORDS FROM THE SAME SERVICE FOR COMPARISON

It has recently been revealed that English users often rationally choose robust passwords for accounts perceived to be of an important value [4], while knowingly choose weak passwords for unimportant accounts [5]. Our survey on Chinese users also reports similar findings (see Fig. 13 of the main text). Since accounts of the same service generally would have the same level of value for users, we divide datasets into three pairs according to their types of service (i.e., Tianya vs. Rockyou, Dodonew vs. Yahoo, and CSDN vs. Phpbp) for fairer strength comparison. We emphasize that, there is little sense if one compares passwords from Dodonew (i.e., an e-commerce site) with passwords from Phpbp (i.e., a low-value programmer forum). Even if Dodonew passwords are stronger than Phpbp passwords, it can never suggest that Chinese passwords are more secure than English passwords, because there is a potential that Dodonew passwords will be weaker than Yahoo e-commerce passwords. However, if we pair password according to their service type, it does reasonably suggest that, for a same/similar service, Chinese passwords will be more secure than English passwords. In this way, we have obtained consistent results for these three
different types of services (see Section V-A in the main text): Chinese passwords are more susceptible to online guessing attacks, while English ones are more vulnerable to offline guessing attacks. As far as we know, we for the first time show the necessity of pairing two datasets on the basis of their service when comparing password security from two different language groups.

APPENDIX E
DETAILED INFORMATION ABOUT THE INSERTION PROCESS OF NAME AND DATE SEGMENTS

Note that each segment in the L- and D-segment dictionaries is associated with a frequency. As there may have already been some Pinyin names in the original L-segment dictionary and the total frequency of these names (denoted by \( n_1 \)) largely reflects the tendency that users insert Pinyin names into their passwords, we insert a name (its frequency denoted by \( f_1 \)) from the “Pinyin name any” dictionary and the L-segment dictionary only if: (1) it is not in the original L-segment dictionary and (2) \( \frac{n_1}{f_1} \geq 1 \), where \( n_2 \) is the total frequency of names that falls into the intersection of the “Pinyin name any” dictionary and the L-segment dictionary. In this way, we manage to only insert a few most frequent ones from an ocean of 2.4M unique names of our name dictionary.

On the other hand, as there are only 27.2K items in our six-digit date dictionary, we take all of them into account. More specifically, for any six-digit date that is not in the original D-segment dictionary, we first associate it with a frequency 1 and then insert it into the D-segment dictionary.

To facilitate the readers reproducing our experiments, here we also provide the detailed password guess generation procedure of our improved PCFG attack in Algorithm 1.

APPENDIX F
COVERAGE OF NAME SEGMENTS IN TEST SETS

As we have shown in Sec. 5.1 of the main text, the training set (i.e., Duowan) is able to well cover the name segments in the test set (i.e., Tianya) and thus the addition of some extra names would be of limited yields. This observation also holds for the other eight test sets and the detailed results are summarized in Table F.1, where “Duowan1M” is Duowan 1M for short and “PY name” is Pinyin name for short. The fraction of L-segments in the test set y that can be covered by the set x is denoted by CoL(x).

Table F.1 shows CoL(x) is at least 11 times larger than CoL(Pinyin_name) or CoL(x), and CoL(Pinyin_name)/CoL(x) is at least 1.9 times larger than CoL(Pinyin_name) / CoL(x), no matter x = Duowan or Duowan 1M. As a result, adding extra

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Tianya</th>
<th>%/R</th>
<th>Dodonew</th>
<th>%/I</th>
<th>178</th>
<th>%/J</th>
<th>CSDN</th>
<th>Avg. Chinese</th>
<th>Rockyou</th>
<th>Yahoo</th>
<th>Phpbb</th>
<th>Avg. English</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>63.77%</td>
<td>59.62%</td>
<td>30.76%</td>
<td>48.07%</td>
<td>45.01%</td>
<td>52.84%</td>
<td>52.93%</td>
<td>15.94%</td>
<td>5.89%</td>
<td>12.06%</td>
<td>15.77%</td>
<td></td>
</tr>
<tr>
<td>LD</td>
<td>14.71%</td>
<td>17.98%</td>
<td>43.50%</td>
<td>26.14%</td>
<td>23.07%</td>
<td>27.70%</td>
<td>27.70%</td>
<td>19.14%</td>
<td>27.70%</td>
<td>15.77%</td>
<td>15.77%</td>
<td></td>
</tr>
<tr>
<td>LDLD</td>
<td>78.48%</td>
<td>77.60%</td>
<td>74.25%</td>
<td>79.20%</td>
<td>71.15%</td>
<td>76.81%</td>
<td>76.66%</td>
<td>43.64%</td>
<td>44.16%</td>
<td>31.20%</td>
<td>43.55%</td>
<td></td>
</tr>
</tbody>
</table>

| Sum of top2 | 41.12% | 3.91% | 7.53% | 6.25% | 5.88% | 5.83% | 5.25% | 2.54% | 5.31% | 2.03% | 2.53% |
| Sum of top10 | 85.82% | 83.52% | 86.56% | 88.41% | 82.97% | 86.51% | 85.45% | 51.64% | 58.37% | 40.31% | 51.64% |

<table>
<thead>
<tr>
<th>Patterns</th>
<th>Tianya</th>
<th>%/R</th>
<th>Dodonew</th>
<th>%/I</th>
<th>178</th>
<th>%/J</th>
<th>CSDN</th>
<th>Avg. Chinese</th>
<th>Rockyou</th>
<th>Yahoo</th>
<th>Phpbb</th>
<th>Avg. English</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>63.77%</td>
<td>59.62%</td>
<td>30.76%</td>
<td>48.07%</td>
<td>45.01%</td>
<td>52.84%</td>
<td>52.93%</td>
<td>15.94%</td>
<td>5.89%</td>
<td>12.06%</td>
<td>15.77%</td>
<td></td>
</tr>
<tr>
<td>LD</td>
<td>14.71%</td>
<td>17.98%</td>
<td>43.50%</td>
<td>26.14%</td>
<td>23.07%</td>
<td>27.70%</td>
<td>27.70%</td>
<td>19.14%</td>
<td>27.70%</td>
<td>15.77%</td>
<td>15.77%</td>
<td></td>
</tr>
<tr>
<td>LDLD</td>
<td>78.48%</td>
<td>77.60%</td>
<td>74.25%</td>
<td>79.20%</td>
<td>71.15%</td>
<td>76.81%</td>
<td>76.66%</td>
<td>43.64%</td>
<td>44.16%</td>
<td>31.20%</td>
<td>43.55%</td>
<td></td>
</tr>
</tbody>
</table>

Algorithm 1: Improved PCFG-based guesses generation

Input: A training set S; A name list nameList; A date list dateList; A parameter k indicating the desired size of the password guess list that will be generated (e.g., \( k = 10^7 \))

Output: A password guess list L with the k highest ranked items

Training (with special attention to monotonically long passwords):

1. for segment ∈ \( \text{splitToSegments} \) do
   2. segmentSet.insert(segment)
3. baseStructure ← baseStructure
4. if monotonicallyLong(baseStructure) then
   5. transformStructureSet.insert(baseStructure)
6. baseStructure ← convertToShort(baseStructure)
7. baseStructureSet.insert(baseStructure)
8. trainingSet.insert(password)

Append name and date lists to the learned segment list

11. for name ∈ nameList do
12.   correctedCount = totalOverlapNameInSegmentSet
13.   nameList.get(name)/totalOverlapNameInNameList
14. if name ∈ segmentSet and correctedCount ≥ 1 then
15.   segmentSet.insert(name, correctedCount)

16. for date ∈ dateList do
17.   if date ∈ segmentSet then
18.     segmentSet.insert(date)

19. function guess:calculateProbability()
20.   guess:probability ← baseStructureSet.getProbability(guess, baseStructure)
21.   if monotonicallyLong(guess, baseStructure) then
22.     baseStructure ← guess, baseStructure
23.     guess:probability ← guess:probability * transformStructureSet.getProbability(baseStructure)
24.     guess:baseStructure ← convertToShort(baseStructure)
25.   for segment ∈ \( \text{splitToSegments} \), guess:password do
26.     guess:probability ← guess:probability * segmentSet.getProbability(segment)
27.   initialize heap:
28.   for baseStructure ∈ baseStructureSet do
29.     segmentType = baseStructure.segmentTypeSet
30.     guess:password ← segmentSet.getFirstSegment(segmentType)
31.     guess:calculateProbability()
32.     guess:segmentChangedPosition ← 1
33.     heap.insert(guess)
34. while guessCount ≤ k do
35.   guess ← heap.pop()
36.   guess:count++
37.   for i ← guess:segmentChangedPosition to guess:baseStructure.length do
38.     if guess:New.password = null then
39.       continue
40.     guess:New.segmentChangedPosition ← i
names into the PCFG L-segments when training is of limited yields. Note that, this does not contradict our observation that Pinyin names are prevalent in Chinese passwords and actually, this does suggest that when training set is selected properly, the name segments in passwords can be well covered. Still, when there is no proper training set available, our improved attack would demonstrate its advantages. Moreover, although our improved PCFG-based algorithm might not be optimal, its cracking results represent a new benchmark that any future algorithm should aim to decisively clear.

**APPENDIX G**

**A SUBTLETY ABOUT GOOD-TURING SMOOTHING ON PASSWORD CRACKING**

There is a subtlety to be noted when implementing the Good-Turing (GT) smoothing technique. We denote \( f \) to be the frequency of an event, and \( N_f \) to be the frequency of frequency \( f \). According to the basic GT smoothing formula, the probability of a string \( "c_1c_2\cdots c_i" \) in a Markov model of order \( n \) is denoted by

\[
P("c_1c_2\cdots c_{i-1}c_i") = \prod_{i=1}^{l} P("c_1c_{i-1}c_{i-2}\cdots c_1"),
\]

where the individual probabilities in the product are computed empirically by using the training sets. More specifically, each empirical probability is given by

\[
P("c_i|c_{i-1}\cdots c_1") = \frac{S(count(c_{i-1}\cdots c_1c_i))}{\sum_{c\in\Sigma} S(count(c_{i-1}\cdots c_1c))},
\]

where the alphabet \( \Sigma \) includes 10 printable numbers on the keyboard plus one special end-symbol (i.e., \( .g. \)) that denotes the end of a password, and \( S(\cdot) \) is defined as:

\[
S(f) = (f + 1) \frac{N_{f+1}}{N_f},
\]

It can be confirmed that this kind of smoothing works well when \( f \) is small, yet it fails for passwords with a high frequency because the estimates for \( S(f) \) are not smooth. For instance, \( 12345 \) is the most common 5-character string in the Rockyou dataset and occurs \( f = 490,044 \) times. Since there is no 5-character string that occurs 490,045 times, \( N_{123455} \) will be zero, implying the basic GT estimator will give a probability 0 for \( P("123455") \). A similar problem regarding the smoothing of frequency of passwords has been identified in [10].

There have been various improvements suggested in linguistics to cope this problem, among which is Gale and Hill’s “simple Good-Turing smoothing” [11]. This improvement is famous for its simplicity and accuracy. This improvement (denoted by SGT) takes two steps of smoothing. Firstly, SGT performs a smoothing for \( N_f \):

\[
SN(f) = \begin{cases} 
N(1) & \text{if } f = 1 \\
2N(f) & \text{if } 1 < f < \max(f) \\
2N(f) & \text{if } f = \max(f)
\end{cases}
\]

where \( f^+ \) and \( f^- \) stand for the next-largest and next-smallest values of \( f \) for which \( N_f > 0 \). Then, SGT performs a linear regression for all values \( SN(f) \) and obtains a Zipf distribution: \( Z(f) = C \cdot (f)^s \), where \( C \) and \( s \) are constants result from regression. Finally, SGT conducts a second smoothing by replacing the raw count \( N_f \) from Eq.3 with \( Z(f) \):

\[
S(f) = \begin{cases} 
(f + 1) \frac{N_{f+1}}{N_f} & \text{if } 0 \leq f < f_0 \\
(f + 1) \frac{Z(f + 1)}{Z(f)} & \text{if } f_0 \leq f
\end{cases}
\]

In 2014, Ma et al. [12] introduced GT smoothing into Markov-based attacks to facilitate more accurate generation of password guesses, yet little attention has been paid to the unsoundness of GT for high frequency events as illustrated above. To the best of our knowledge, we for the first time well explicate the combination uses of GT and SGT in Markov-based password cracking.

**APPENDIX H**

**MARKOV-CHAIN-BASED CRACKING RESULTS**

The Markov-Chain-based cracking results for attacking Scenario #1 have been included in Fig.5 of the main text, and the experiments for the four remaining scenarios are depicted in Figs. 1, 2, 3 and 4, respectively. From these figures, one can see that for both Chinese and English test sets: (1) At large guesses (i.e., no less than \( 2 \times 10^6 \)), order-4 markov-chain evidently performs better than the other two orders, while at small guesses (i.e., less than \( 10^6 \)) the larger the order, the better the performance will be; (2) There is no much difference in performance between Laplace Smoothing and Good-Turing Smoothing at small guesses, while the advantage of Laplace Smoothing gets larger as the guess number increases; (3) End-symbol-based normalization always performs better than the distribution-based approach, while at small guesses its advantages will be more obvious; (4) Backoff ordering and Smoothing (with distribution-based normalization) performs marginally better than the other four scenarios, yet we find that it is over 10 times slower and consumes over 15 times more memory.

Our cracking results suggest that, at large guesses, the attacks preferring order-4, Laplace Smoothing and end-symbol-based normalization perform the best among all the series of Markov-chain-based attacks; at small guesses (e.g., less than

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Duowan</td>
<td>17.26%</td>
<td>80.05%</td>
<td>5.90%</td>
<td>65.67%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
</tr>
<tr>
<td>Duowan_test</td>
<td>18.07%</td>
<td>75.03%</td>
<td>13.46%</td>
<td>65.67%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
<td>6.68%</td>
<td>61.57%</td>
</tr>
</tbody>
</table>
10^6), the attacks preferring order-5, Laplace Smoothing and end-symbol-based normalization perform the best among all the series of Markov-chain-based attacks.

It is worth noting that, the “reversal principle” also applies in all the markov-chain-based experiments. For example, in order-4 markov-chain-based experiments (see Fig.2(b) and Fig.2(e)), we can see that, when the guess number is below about 7000, Chinese web passwords are generally much weaker than their English counterparts. For example, at 1000 guesses, the success rate against Tianya, Dodonew and CSDN is 11.8%, 6.3% and 11.6%, respectively, while their English counterparts (i.e., Rockyou, Yahoo and Phpbb) is merely 8.1%, 4.3% and 7.1%, respectively. However, when the guess number is allowed to be over 10^4, Chinese web passwords are generally stronger than their English counterparts. For example, at 1 million guesses, the success rate against Tianya, Dodonew and CSDN is 38.2%, 20.4% and 25.4%, respectively, while their English counterparts is 38.6%, 24.8% and 32.3%, respectively.

**APPENDIX I**

**SOME USEFUL IMPLICATIONS**

We now discuss some useful implications that our findings in Sec. III to Sec V are likely to carry.

**A. Implications for password creation policies**

It is interesting to see that, 97.83% passwords in CSDN are of length 8+ CSDN enforces a minimum length-8 policy (see Fig. 18 of the main text), while Dodonew enforces no apparent rule (i.e., neither minimum length nor character set requirement) as is evident from Table III of the main text. However, Fig. 1 to Fig. 4 indicate that, given any guess number below 10^7, passwords from CSDN is significantly weaker than passwords from Dodonew. A plausible reason is that Dodonew provides e-commerce services, and most users perceive it as important. As a result, users rationally choose more complex passwords for it. As for CSDN, since it is only a technology community, users knowingly choose weak passwords for it.

In 2012, Bonneau [13] cast doubt on the hypothesis that users will rationally select more secure passwords to protect their more important accounts. In 2013 Egelman et al. [5] initiated a field study involving 51 students and confirmed this hypothesis. In 2014, Stobert and Biddle [4] interviewed 27 participants to investigate user behaviour in managing passwords, and their results also corroborate this hypothesis. As far as we know, here we for the first time provide a large-scale empirical evidence (i.e., on the basis of 6.43 million CSDN passwords and 16.26 million Dodonew passwords) that supports this hypothesis.

We also note that though the overall security of Dodonew passwords are higher than passwords from the five other Chinese sites, many popular passwords dwelling in Dodonew also appear in other less sensitive sites (see Table III of the main text). This might be due to that users inadvertently choose popular passwords (because they do not what are the passwords that other users have chosen [14]) and that many users reuse the same password across multiple sites (see Figs 2 and 3 in our user survey). What’s even more dangerous is that many users fail to recognize different categories of accounts, because achieving this goal is not easy [15]. Further considering the “over-constrained nature of authentication” on the Web [16] and the “finite-effort user” [17], we suggest that when designing password creation policies, instead of merely insisting on stringent rules, administrators should put more efforts on helping users gain more accurate perceptions of the importance of the accounts to be protected and on guiding users towards better ability of recognizing different categories of accounts. Both efforts would help enhance user internal impetus and are essential for common users to responsibly allocating (i.e., selecting one candidate from their limited pool of passwords memorized [4], [15]) passwords.

In addition, the “reversal principle” revealed in our work
shows that Chinese passwords are more vulnerable to online guessing attacks. This is highly due to the fact that top popular Chinese passwords are more concentrated (see Table III of the main text). Thus, a special blacklist that includes a moderate number (e.g., 50K as suggested in [18]) of most common Chinese passwords would be very helpful for Chinese users to avoid trivial online guessing attacks. Such a blacklist can be learned from various leaked Chinese datasets (see one list in http://t.cn/RG88tvF). Any password falling into this list shall be deemed weak. However, it is well known that if some popular passwords are banned, new popular ones will arise. These new popular passwords may be complex and subtle to detect. Hence, whenever possible, besides password creation policies (rules), password strength meters shall be further employed by critical services (e.g., email and e-commerce) to detect weak user-chosen passwords.

B. Implications for password strength meters

Password strength meters (PSMs) are generally used along with password creation policies (rules) to nudge users towards better passwords. While password policies tell a user what constitute a good (or an acceptable) password, PSMs feedback to a user how her submitted password performs (weak or not?) in the face of attackers. Several studies (e.g., [19], [20]) have shown that PSMs of high-profile service providers are highly inaccurate and inconsistent at assessing the security of weak Chinese passwords. Failing to provide reliable feedbacks to users would have great negative effects such as user confusion, frustration and distrust.

It is suggested that PSMs “can simplify challenges by limiting their primary goal only to detecting weak passwords, instead of trying to distinguish a good, very good, or good password” [20]. It follows that the essential step of a PSM would be to identify the characteristics of weak passwords. From our findings in Section IV and Section V of the main text, it is evident that for passwords of Chinese users, the incorporation of long Pinyin words or full/family names is a compelling evidence for a “weak” decision. Other signs of weak passwords are the incorporation of birthdays and simple patterns like repetition and palindrome.

Figs. 3 and 4 of the main text reveals that Chinese users also tend to reuse passwords, and this issue is as severe as English users. This means that when Chinese users construct passwords for a new service, they actually reuse or simply modify an existing password, but do not build passwords from scratch by concatenating segments of words, digits and/or symbols (i.e., the PCFG-based approach [21]) or by combining Markov n-grams (i.e., the Markov-based approach [22]). This outlines the need for these existing PSMs (e.g., [21], [22]) to better capture users’ reuse behaviors.

C. Implications for password cracking

Based on our extensive cracking experiments, we find that, as compared to Markov-based attacks, PCFG-based ones are simpler to implement (in terms of both computation and memory cost), and they perform equally well (even better, see Fig. 1 to Fig. 4) when the guess number is small (e.g., $10^3$). For large guess numbers, order-4 Markov-based attacks are the best choices over order-3 and 5. As as as we know, these observations have not been previously elucidated (e.g., [12], [23]). Note that we have only shown the Markov-based results when the guess number is below $10^7$, there is a potential that order-3 Markov-based attacks will outperform order-4 and 5 ones at larger guess numbers (e.g., $10^{14}$).

REFERENCES FOR APPENDIX

Fig. 3. Markov-Chain-based attacks on different groups of datasets (using Good-Turing Smoothing and Distribution-based Normalization). Attacks (a)–(c) use 1 million Duowan passwords as the training set, while attacks (d)–(f) use 1 million Rockyou as the training set.

Fig. 4. Markov-Chain-based attacks on different groups of datasets (using Backoff smoothing and Distribution-based Normalization). Attack (a) uses 1 million Duowan passwords as the training set, while attack (b) uses 1 million Rockyou as the training set.


