Birthday, Name and Bifacial-security: Understanding Passwords of Chinese Web Users

Ding Wang and Ping Wang, *Peking University;* Debiao He, *Wuhan University;* Yuan Tian, *University of Virginia*

https://www.usenix.org/conference/usenixsecurity19/presentation/wang-ding

This paper is included in the Proceedings of the 28th USENIX Security Symposium.

August 14–16, 2019 • Santa Clara, CA, USA

978-1-939133-06-9

Birthday, Name and Bifacial-security: Understanding Passwords of Chinese Web Users

Ding Wang[†*], Ping Wang[†*], Debiao He[§], Yuan Tian[‡]
[†] *Peking University, Beijing 100871, China; {wangdingg, pwang}@pku.edu.cn*
[*]*Key Lab of High-Condence Software Technology (PKU), Ministry of Education, China*
[§]*School of Cyber Science and Engineering, Wuhan University, China; hedebiao@whu.edu.cn*
[‡]*School of Engineering and Applied Science, University of Virginia; yuant@virginia.edu*

Abstract

Much attention has been paid to passwords chosen by English speaking users, yet only a few studies have examined how *non*-English speaking users select passwords. In this paper, we perform *an extensive, empirical analysis* of 73.1 million real-world Chinese web passwords in comparison with 33.2 million English counterparts. We highlight a number of interesting structural and semantic characteristics in Chinese passwords. We further evaluate the security of these passwords by employing two state-of-the-art cracking techniques. In particular, our cracking results reveal the *bifacial-security* nature of Chinese passwords. They are weaker against online guessing attacks (i.e., when the allowed guess number is small, 1$\sim$$10^4$) than English passwords. But out of the remaining Chinese passwords, they are stronger against offline guessing attacks (i.e., when the guess number is large, $>$$10^5$) than their English counterparts. This reconciles two conflicting claims about the strength of Chinese passwords made by Bonneau (IEEE S&P'12) and Li et al. (Usenix Security'14 and IEEE TIFS'16). At 10^7 guesses, the success rate of our improved PCFG-based attack against the Chinese datasets is 33.2%$\sim$49.8%, indicating that our attack can crack 92% to 188% *more* passwords than the state of the art. We also discuss the implications of our findings for password policies, strength meters and cracking.

1 Introduction

Textual passwords are the dominant form of access control in almost every web service today. Although their security pitfalls were revealed as early as four decades ago [39] and various alternative authentication methods (e.g., graphical passwords and multi-factor authentication) have been proposed since then, passwords are still widely used. For one reason, passwords offer many advantages, such as low deployment cost, easy recovery, and remarkable simplicity, which cannot always be offered by other authentication methods [6]. For another reason, there is a lack of effective tools to quantify the less obvious costs of replacing passwords [8] because the marginal gains are often insufficient to make up for the significant transition costs. Furthermore, users also favor passwords. A recent survey on 1,119 US users [49] showed that 58% of the participants prefer passwords as their online login credentials, while only 16% prefer biometrics, and 10% prefer other ways. Thus, passwords are likely to persist in the foreseeable future.

Despite its ubiquity, password authentication is confronted with a challenge [62]: truly random passwords are difficult for users to memorize, while easy-to-remember passwords tend to be highly predictable. To eliminate this notorious "security-usability" dilemma, researchers have put a lot of effort [12,17,36,46,47] into the following two types of studies.

Type-1 research aims at evaluating the strength of a password dataset (distribution) by measuring its statistical properties (e.g., Shannon entropy [10], α-guesswork [7], λ-success-rate [53]) or by gauging its "guessability" [24, 59]. Guessability characterizes the fraction of passwords that, at a given number of guesses, can be cracked by cracking algorithms such as Markov-Chains [36] and probabilistic context-free grammars (PCFG) [58]. As with most of these previous studies, we mainly consider *trawling guessing* [55], while other attacking vectors (e.g., phishing, shoulder-surfing and targeted guessing [56]) are outside of our focus. Hereafter, whenever the term "guessing" is used, it means trawling guessing.

Type-2 research attempts to reduce the use of weak passwords. Two approaches have been mainly utilized: proactive password checking [25, 32] and password strength meter [13, 59]. The former checks the user-selected passwords and only accepts those 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 [17]. Most of today's leading sites employ a combination of these two approaches to prevent users from choosing weak passwords. In this work, though we mainly focus on type-1 research, our findings are also helpful for type-2 research.

Existing work (e.g., [14,17,27,37,42]) mainly focuses on passwords chosen by English speaking users. Relatively little attention has been paid to the characteristics and strength of passwords chosen by those who speak other native languages. For instance, "woaini1314" is currently deemed "Strong" by password strength meters (PSMs) of many leading services like AOL, Google, IEEE, and Sina weibo. However, this password is highly popular and prone to guessing [56]: "woaini" is a Chinese Pinyin phrase that means "I love you", and "1314" has a similar pronunciation of "for ever" in Chinese. Failing to catch this would overlook the weaknesses of Chinese passwords, thus posing high risks to the corresponding web accounts.

1.1 Motivations

There have been 802 million Chinese netizens by June, 2018 [1], which account for over 20% (and also the largest fraction) of the world's Internet population. However, to the best of our knowledge, there has been no satisfactory answer to the key questions: *(1) Are there structural or semantic characteristics that differentiate Chinese passwords from English ones? (2) How will Chinese passwords perform against the foremost attacks? (3) Are they weaker or stronger than English ones?* It is imperative to address these questions to provide both security engineers and Chinese users with necessary security guidance. For instance, if the answer to the first question is affirmative, then it indicates that *the password policies (e.g., length-8^+ [25] and 2Class12 [44]) and strength meters (e.g., RNN-PSM [38] and Zxcvbn [59]) originally designed for English speaking users cannot be readily applied to Chinese speaking users.*

A few password studies (e.g., [30,36,52,53,56]) have employed some Chinese datasets, yet they mainly deal with the effectiveness of various probabilistic cracking models. Relatively little attention has been given to the above three questions. As far as we know, Li et al.'s work [26, 34] may be the closest to this paper, but our work differs from it in several aspects. First, we explore a number of fundamental characteristics not covered in [26, 34], such as the extent of language dependence, length distribution, frequency distribution and various semantics. Second, our improved PCFG-based algorithm can achieve success rates from 29.41% to 39.47% at *just* 10^7 guesses, while the best success rate of their improved PCFG-based algorithm is only 17.3% at 10^{10} guesses (i.e., significantly underestimate attackers). Third, based on more comprehensive experiments, we outline the need for pairing passwords in terms of site service type when comparing password strength, which is overlooked by Li et al.'s [26, 34] and Bonneau's [7] work. Fourth, as shown in Sec. 3.2, two of Li et al.'s five Chinese datasets are improperly pre-processed when they perform data cleaning,[1] which impairs their results.

1.2 Contributions

We perform a large-scale empirical analysis by leveraging 73.1 million passwords from six popular Chinese sites and 33.2 million passwords from three English sites. Particularly, we seek for fundamental properties of user-generated passwords and systematically measure their structural patterns, semantic characteristics and strength. In summary, we make the following key contributions:

- **An empirical analysis**. By leveraging 73.1 million real-life Chinese passwords, *for the first time*, we: (1) provide a *quantitative* measurement of to what extent user passwords are influenced by their native language; (2) *systematically* explore the common semantics (e.g., date, name, place and phone #) in passwords; and (3) show that passwords of these two distinct user groups follow quite similar Zipf frequency distributions, despite being created under diversified password policies.

- **A reversal principle**. We employ two state-of-the-art password-cracking algorithms (i.e., PCFG-based and Markov-based [36]) to measure the strength of Chinese web passwords. We also improve the PCFG-based algorithm to more accurately capture passwords that are of a monotonically long structure (e.g., "1qa2ws3ed"). At 10^7 guesses, our algorithm can crack 92% to 188% more passwords than the best results in [34]. Particularly, we reveal a "reversal principle", i.e. the *bifacial-security* nature of Chinese passwords: when the guess number allowed is small, they 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 [7,34].

- **Some insights**. We highlight some insights for password policies, strength meters and cracking. We provide a large-scale empirical evidence that supports the hypothesis raised in the HCI community [17, 46]: users self-reported to rationally choose stronger passwords for accounts associated with a higher value, and knowingly select weaker passwords for a lower-value service even if the latter imposes a stricter policy. Our methodological approaches would also be useful for analyzing passwords of other non-English speaking users.

[1] We reported this issue to the authors of [26, 34], they have acknowledged it. As their journal paper [26] is technically a verbatim of their conference version [34], we mainly use [34] for discussion.

2 Related work

In this section, we briefly review prior research on password characteristics and security.

2.1 Password characteristics

Basic statistics. In 1979, Morris and Thompson [39] analyzed a corpus of 3,000 passwords. They reported that 71% of the passwords are no more than 6 characters long and 14% of the passwords are non-alphanumeric characters. In 1990, Klein [32] collected 13,797 computer accounts from his friends and acquaintances around US and UK. They 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% of the collected accounts and 52% of the cracked passwords are shorter than 6 characters long. In 2004, Yan et al. [62] found that passwords are likely to be dictionary words since users have difficulty in memorizing random strings. On average, the password length in their user study (288 participants) is 7∼8.

In 2012, Bonneau [7] conducted a systematic analysis of 70 million Yahoo private 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). They found that even seemingly distant language communities choose the same weak passwords. This research was recently reproduced in [3] by using differential privacy techniques. Particularly, Chinese passwords are found among the most difficult ones to crack [7]. In 2014, however, Li et al. [34] 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 speaking users who like to use letters to create passwords. However, as an elementary defect, two of their Chinese datasets have not been cleaned properly (see Section 3.2), which might lead to inaccurate measures and biased comparisons. More importantly, several critical password properties (such as length distributions, frequency distributions and semantics) remain to be explored.

In 2014, Ma et al. [36] 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 semantics (e.g., no information is provided about the role of Pinyins, names or dates). In 2017, Pearman et al. [42] reported on an in situ examination of 4057 passwords from 154 English-speaking users over an average of 147 days. They found that the average password is composed of 2.77 character classes and is of length 9.92 characters, including 5.91 lowercase letters, 2.70 digits, 0.84 uppercase letters, and 0.46 symbols.

Semantic patterns. In 1989, Riddle et al. [43] found that birth dates, personal names, nicknames and celebrity names are popular in user-generated passwords. In 2004, Brown et al. [9] confirmed this by conducting a thorough survey that involved 218 participants and 1,783 passwords. They reported that the most frequent entity in passwords is the self (67%), followed by relatives (7%), lovers and friends; Also, names (32%) were found to be the most common information used, followed by dates (7%). Veras et al. [51] examined the 32M RockYou dataset by employing visualization techniques and observed that 15% 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. [34] showed that Chinese users tend to insert Pinyins and dates into their passwords. However, many other important semantic patterns (e.g., Pinyin name and mobile number) are left unexplored. In addition, we improve upon the processes of data cleaning (see Sec. 3.2) and tuning of cracking algorithms (see Sec. 4.1) to advance beyond Li et al.'s measurement of the strength of Chinese passwords. In 2015, Ji et al. [30] noted that user-IDs and emails have a great impact on password security. For instance, 53% of Dodonew passwords can be guessed by using user-IDs within an average of 706 guesses. This motivates us to investigate to what extent the Pinyin names and Chinese-style dates impact the security of Chinese passwords. In 2018, AlSabah et al. [2] studied 79,760 passwords leaked from the Qatar National Bank, customers of which are mainly Middle Easterners. They observed that over 30% of passwords contain names, over 5% use a 2-digit birth year, and 4% include their own phone number in whole as part of their password.

2.2 Password security

A crucial password research subject is password strength. Instead of using brute-force attacks, earlier works (e.g., [32, 43]) use a combination of ad hoc dictionaries and mangling rules, in order to model the common password generation practice and see whether user passwords can be successfully rebuilt in a period of time. This technique has given rise to automated tools like John the Ripper (JTR), hashcat and L0phtCrack.

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 a bonus for special checks. Florencio and Herley [19], and Egelman et al. [17] improved this approach by adding the size of the alphabet into the calculation and called the resulting value $log_2((alpha.size)^{pass.len})$ 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. [57]. 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 by using PCFG, and then instantiates the derived grammars by using string segments from external input dictionaries to generate guesses in decreasing probability order [58]. This PCFG-based cracking approach is able to crack 28% to 129% more passwords than JTR when allowed the same guess number. It is considered as a leading password cracking technique and used in a number of recent works [36, 56].

Differing from the PCFG-based approach, Narayanan and Shmatikov [40] introduced the Markov-Chain theory for assigning probabilities to letter segments, which substantially reduces the password search space. This approach was tested in an experiment against 142 real user passwords and could break 68% of them. In 2014, by utilizing various normalization and smoothing techniques from the natural language processing domain, Ma et al. [36] systematically evaluated the Markov-based model. They found it performs significantly better than the PCFG-based model at large guesses (e.g., 2^{30}) in some cases when parameterized appropriately. In this work, we perform extensive experiments by using both models to evaluate the strength of Chinese passwords.

When these password models are coupled with tools (e.g., AUTOFORGE [63]) that can automatically forge valid online login requests from the client side, server-side mechanisms like rate-limiting (see Sec. 5.2.2 of [25]) and password leakage detection [31] become necessary. However, in reality, few sites have implemented proper countermeasures to thwart online guessing. Among the 182 sites in the Alexa Top 500 sites in the US that Lu et al. [35] were able to examine, 131 sites (72%) allow frequent unsuccessful login attempts, and another 28 sites (15%) can be easily locked out, leading to denial of service attacks. This further suggests the necessity of our work—understanding the strength of Chinese passwords against online guessing.

3 Characteristics of Chinese passwords

We now investigate Chinese password characteristics, most of which are underexplored. In addition, we discuss weaknesses in previous major studies [26, 34].

3.1 Dataset and ethics consideration

Our empirical analysis employs six password datasets from Chinese websites and three password datasets from English websites. In total, these nine datasets consist of 106.3 million real-life passwords. As summarized in Table 1, these nine datasets are different in terms of service, language, culture, and size. The role of each dataset will be specified in Sec. 4 when performing strength comparison. They were hacked and made public on the Internet between 2009 and 2012, and may be a bit old. However, they can represent current passwords due to two reasons. First, Bonneau has shown that "passwords have changed only marginally since then (1990)" [7]. Second, the password ecosystem evolves very slowly. A number of recent researches (see [21, 24, 55]) reveal that password guidance and practices implemented on leading sites have seldom changed over time.

We realize that though publicly available and widely used in the literature [36, 52, 56], these datasets are private data. Thus, we only report the aggregated statistical information, and treat each individual account as confidential so 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 is both beneficial for the academic community to understand password choices of Chinese netizens and for security administrators to secure user accounts. As our datasets are all publicly available, the results in this work are reproducible.

3.2 Data cleaning

We note that some original datasets (e.g., Rockyou and Tianya) include *un*-necessary headers, descriptions, footnotes, password strings with len>100, etc. Thus, before any exploration, we first launch data cleaning. We remove email addresses and user names from the original data. As with [36], we also remove strings that include symbols beyond the 95 printable ASCII characters. We further remove strings with len>30, because after manually scrutinizing the original datasets, we find that these long strings do *not* seem to be generated by users, but more likely by password managers or simply junk information. Moreover, such unusually long passwords are often beyond the scope of attackers who care about cracking efficiency [4]. In all, the fraction of excluded passwords is *negligible* (see the last column but two in Table 1), yet this cleaning step unifies the input of cracking algorithms and simplifies the later data processing.

We find that either Tianya or 7k7k has been contaminated: there is a *non-negligible* overlap between the Tianya dataset and 7k7k dataset (i.e., 40.85% of 7k7k and 24.62% of Tianya). More specifically, we were first puzzled by the fact that the password "`111222tianya`"

Table 1: Data cleaning of the password datasets leaked from nine web services ("PWs" stands for passwords).

Dataset	Web service	Language	Leaked Time	Original PWs	Miscellany	Length>30	Removed %	After cleaning	Unique PWs
Tianya	Social forum	Chinese	Dec. 2011	31,761,424	860,178	5	2.71%	30,901,241	12,898,437
7k7k	Gaming	Chinese	Dec. 2011	19,138,452	13,705,087	10,078	71.66%*	5,423,287	2,865,573
Dodonew	E-commerce&Gaming	Chinese	Dec. 2011	16,283,140	10,774	13,475	0.15%	16,258,891	10,135,260
178	Gaming	Chinese	Dec. 2011	9,072,966	0	1	0.00%	9,072,965	3,462,283
CSDN	Programmer forum	Chinese	Dec. 2011	6,428,632	355	0	0.01%	6,428,277	4,037,605
Duowan	Gaming	Chinese	Dec. 2011	5,024,764	42,024	10	0.83%	4,982,730	3,119,060
Rockyou	Social forum	English	Dec. 2009	32,603,387	18,377	3140	0.07%	32,581,870	14,326,970
Yahoo	Portal(e.g., E-commerce)	English	July 2012	453,491	10,657	0	2.35%	442,834	342,510
Phpbb	Programmer forum	English	Jan. 2009	255,421	45	3	0.02%	255,373	184,341

*We remove 13M duplicate accounts from 7k7k, because we identify that they are copied from Tianya as we will detail in Section 3.2.

was originally in the top-10 most popular list of both datasets. We manually scrutinize the original datasets (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. In Appendix A, we provide strong evidence that someone has copied these joint accounts from Tianya to 7k7k, but *not* from 7k7k to Tianya as concluded in previous major studies [26, 34].

3.3 Password characteristics

Language dependence. There is a folklore that user-generated passwords are greatly influenced by their native languages, yet so far no large-scale quantitative measurement has ever been given. To fill this gap, we first illustrate the character distributions of the nine 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. 1). 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 the native language of its users is by investigating its letter distribution. Arranged in descending order, the letter distribution of all Chinese passwords is aineohglwuyszxqcdjmbtfrkpv, while this distribution for all English passwords is aeionrlstmcdyhubkgpjvfwzxq. While some letters (e.g., 'a', 'e' and 'i') occur frequently in both groups, some letters (e.g., 'q' and 'r') 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 frequency of general language may be somewhat different from the letter frequency of passwords. *To what extent do they differ?* According to Huang et al.'s work [28], the letter distribution of Chinese language (i.e., written Chinese texts like literary work, newspapers and academic papers), when converted into Chinese Pinyin, is inauhegoyszdjmxwqbctlpfrkv. 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 names 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 create their passwords.

A similar observation holds for passwords of English speaking users. The letter distribution of English language (i.e., etaoinshrdlcumwfgypbvkjxqz) is from www.cryptograms.org/letter-frequencies.php. For example, 't' is common in English texts, but not so common in English passwords. A plausible reason may be that 't' is used in popular words like the, it, this, that, at, to, while such words are rare in passwords.

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 sequences (in descending order) between two password datasets (as well as languages). The results are summarized in Table 2. "Pinyin_fullname" is a dictionary consisting of 2,426,841 unique Chinese full names (e.g., wanglei and zhangwei), "Pinyin_word" is a dictionary consisting of 127,878 unique Chinese words (e.g., chang and cheng), and these two dictionaries are detailed in Appendix B. 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 inauh to aniuh is 3, which is equal to that of aniuh to inauh.

As shown in Table 2, 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 2). The latter is less expected. All this indicates that passwords from different languages are intrinsically different from each other in letter distributions, and that passwords are close to their native language yet the distinction is still significant (measurable).

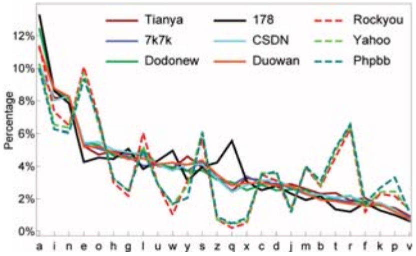


Figure 1: Letter distributions of passwords.

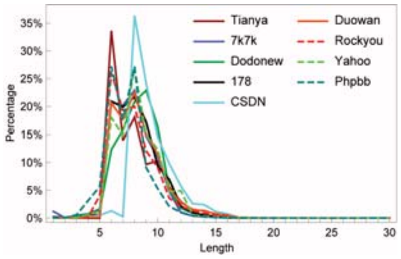


Figure 2: Length distributions of passwords.

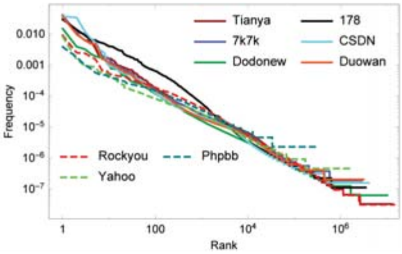


Figure 3: Freq. distributions of passwords.

Table 2: Inversion number of the letter distributions (in descending order) between two datasets.

	Tianya	7k7k	178	CSDN	Dodonew	Duowan	All Chinese PWs	Chinese language	Pinyin fullname	Pinyin word	Rockyou	Yahoo	Phpbb	All English PWs	English language
Tianya	0	15	22	42	15	17	14	40	32	37	100	100	113	100	99
7k7k	15	0	23	31	14	10	13	41	39	38	105	101	112	105	96
Dodonew	22	23	0	42	21	15	12	52	40	49	94	92	105	94	99
178	42	31	42	0	41	35	32	56	48	47	134	130	141	134	125
CSDN	15	14	21	41	0	12	15	45	39	42	95	95	106	95	96
Duowan	17	10	15	35	12	0	9	49	39	44	99	97	110	99	98
All_Chinese_PWs	14	13	12	32	15	**9**	0	44	34	43	104	102	115	104	101
Chinese_language	40	41	52	56	45	49	44	0	38	27	118	114	123	118	113
Pinyin_fullname	32	39	40	48	39	39	34	38	0	31	124	122	135	124	123
Pinyin_word	37	38	49	47	42	44	43	27	31	0	115	113	124	115	112
Rockyou	100	105	94	134	95	99	104	118	124	115	0	12	23	0	47
Yahoo	100	101	92	130	95	97	102	114	122	113	12	0	15	12	39
Phpbb	113	112	105	141	106	110	115	123	135	124	23	15	0	23	44
All_English_PWs	100	105	94	134	95	99	104	118	124	115	**0**	12	23	0	47
English_language	99	96	99	125	96	98	101	113	123	112	47	39	44	47	0

Note that, among all Chinese datasets, Duowan has the least inversion number (i.e., 9 in dark gray) with the dataset "All_Chinese_PWs". This indicates that *Duowan passwords are likely to best represent general Chinese web passwords, and thus Duowan will be selected as the training set for attacking other Chinese datasets (see Sec 5)*. For a similar reason, Rockyou will be selected as the training set when attacking English passwords.

Length distribution. Fig. 2 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 and 8 take the lead. Merely passwords with lengths of 6 to 10 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. Very few users prefer passwords longer than 15 characters. Notably, people seem to prefer even lengths over odd ones. Another interesting observation is that, CSDN exhibits only one peak in its length distribution curve and has many fewer passwords (i.e., only 2.16%) with length<8. This might be due to the password policy that requires the length to be no shorter than 8 on this site.

Frequency distribution. Fig. 3 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. Then, each individual password will be associated with a frequency f_r, and its rank in the frequency table is denoted by r. 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 accords with the Zipf's law [53]: f_r and r follow a relationship of the type $f_r = C \cdot r^s - C \cdot (r-1)^s \approx C \cdot s \cdot r^{s-1}$, where $C \in$[0.01, 0.06] and $s \in$[0.15, 0.40] are constants. Particularly, $1-s$ is the absolute value of the Zipf linear regression line's slope. The Zipf theory indicates that *the popularity of passwords decreases polynomially with the increase of their rank*. This further implies that a few passwords are overly popular (explaining why online guessing [56] can be effective, even if security mechanisms like rate-limiting and suspicious login detection [16] are implemented at the server), while the least frequent passwords are very sparsely scattered in the password space (explaining why offline guessing attackers need to consider cost-effectiveness [4] and weigh when to stop).

Top popular passwords. Table 3 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 due to its password policy that requires passwords to be of length 8^+ (see Fig. 2). "111111" follows on the heel. Other popular Chinese passwords include "123123", "123321" and "123456789", all composed of digits and in simple patterns such as repetition and palindrome. Love also shows its magic power: "5201314", which has a similar pronunciation of "I love you forever and ever" in Chinese,[2] appears in the top-10 lists of four Chinese

[2] https://ninchanese.com/blog/2016/05/20/520-chinese-love-word-number/

Table 3: Top-10 most popular passwords of each dataset.

Rank	Tianya	7k7k	Dodonew	178	CSDN	Duowan	Rockyou	Yahoo	Phpbb
1	123456	123456	123456	123456	123456789	123456	123456	123456	123456
2	111111	0	a123456	111111	12345678	111111	12345	password	password
3	000000	111111	123456789	zz12369	11111111	123456789	123456789	welcome	phpbb
4	123456789	123456789	111111	qiulaobai	dearbook	123123	password	ninja	qwerty
5	123123	123123	**5201314**	123456aa	00000000	000000	**iloveyou**	abc123	12345
6	123321	**5201314**	123123	wmsxie123	123123123	**5201314**	princess	123456789	12345678
7	**5201314**	123	a321654	123123	1234567890	123321	123321	12345678	letmein
8	12345678	12345678	12345	000000	88888888	a123456	rockyou	sunshine	111111
9	666666	12345678	000000	qq66666	111111111	suibian	12345678	princess	1234
10	111222tianya	wangyut2	123456a	w2w2w2	147258369	12345678	abc123	qwerty	123456789
Sum of top-10	2,297,505	440,300	533,285	793,132	670,881	338,012	669,126	4,476	7,135
Total accounts	30,901,241	5,423,287	16,258,891	9,072,965	6,428,277	4,982,730	32,581,870	442,834	255,373
% of top-10	7.43%	8.12%	3.28%	8.74%	10.44%	**6.78%**	2.05%	1.01%	**2.79%**

Table 4: Top-3 structural patterns in two user groups (each % is taken by dividing the corresponding total accounts).

Top-3 patterns	Chinese password datasets						Average of	Top-3 patterns	English password datasets			Average of
in Chinese PWs	Tianya	7k7k	Dodonew	178	CSDN	Duowan	Chinese PWs	in English PWs	Rockyou	Yahoo	Phpbb	English PWs
D(e.g., 123456)	**63.77%**	**59.62%**	**30.76%**	**48.07%**	**45.01%**	**52.84%**	**52.93%**	L(e.g., abcdef)	**41.69%**	33.03%	**50.07%**	**41.59%**
LD(e.g., a12345)	14.71%	17.98%	43.50%	31.12%	26.14%	23.97%	23.72%	LD(e.g., abc123)	27.70%	38.27%	19.14%	28.37%
DL(e.g., 12345a)	4.12%	3.91%	7.55%	6.25%	5.88%	5.83%	5.25%	D(e.g., 123456)	15.94%	5.89%	12.06%	11.30%
Sum of top-3	82.61%	81.51%	81.80%	85.45%	77.03%	82.64%	**81.90%**	Sum of top-3	85.33%	77.19%	81.25%	**81.26%**

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 perhaps euphemistically, "princess"—also show up in top-10 lists of English datasets. Our results confirm the folklore [50] that "back at the dawn of the Web, the most popular password was 12345. Today, it is one digit longer but hardly safer: 123456."

It is interesting to see that only the top-10 most popular ones account for as high as 6.78%∼10.44% of each entire dataset, with Dodonew being the only exception. However, this figure for Dodonew even achieves 3.24%, while the English datasets are all below 2.80%. This indicates that top-popular Chinese passwords are more concentrated than their English counterparts, which is likely to make Chinese passwords more vulnerable to online guessing. This will be confirmed in Sec. 4.1.

Top popular structures. We have seen that digits are popular in top-10 passwords of Chinese datasets. Are they also popular in the whole datasets? We investigate the frequencies of password patterns that involve *digits*, and show the results of the top 3 most frequent ones in the left hand of Table 4. The first column of the table denotes the pattern of a password as in [58] (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 the password "Wanglei123" is ULD). Over 50% of the average Chinese web passwords are only composed of digits, while this value for English datasets is only 11.30%. In contrast to first D then DL, English speaking users prefer the patterns L and LD.

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

Semantics in passwords. As there is little existing work, to gain an insight into the underlying semantic patterns, we have to construct semantic dictionaries from scratch by ourselves. Finally, we construct 22 dictionaries of different semantic categories (see the first column in Table 5). The detailed information about how we construct them is referred to Appendix B. To eliminate ambiguities, we use the "left-most longest match" when matching a password with each item in our dictionaries. Table 5 shows the prevalence of various semantic patterns in passwords. Lots of English speaking users tend to use raw English words as their password building blocks: 25.88% insert a 5^+-letter word into their passwords. Passwords with a 5^+-letter word account for over a third of the total passwords with a 5^+-letter substring. In comparison, fewer Chinese users (2.41%) choose English words to build passwords, yet they prefer Pinyin names (11.50%), especially *full names*.

Particularly, of all the Chinese passwords (22.42%) that include a 5^+-letter substring, more than half

Table 5: Popularity of 22 kinds of semantics in passwords (by matching our 22 semantic dictionaries).*

Semantic dictionary	Tianya	7k7k	Dodonew	178	CSDN	Duowan	Avg Chinese	Rockyou	Yahoo	Phpbb	Avg English
English_word_lower(len $\geq$ 5)	2.08%	2.05%	3.69%	0.83%	3.41%	2.37%	2.41%	**23.54%**	**29.49%**	**24.60%**	**25.88%**
English_firstname(len $\geq$ 5)	1.11%	0.93%	2.23%	0.53%	1.47%	1.19%	1.24%	18.80%	15.21%	9.20%	14.40%
English_lastname(len $\geq$ 5)	2.16%	2.34%	4.48%	1.93%	3.65%	2.77%	2.89%	**20.16%**	**20.82%**	**15.22%**	**18.73%**
English_fullname(len $\geq$ 5)	4.03%	4.30%	6.14%	4.99%	6.58%	5.07%	5.18%	13.05%	11.35%	8.25%	10.88%
English_name_any(len $\geq$ 5)	4.60%	4.65%	6.32%	5.20%	6.87%	5.18%	5.35%	**27.67%**	**26.51%**	**18.71%**	**24.30%**
Pinyin_word_lower(len $\geq$ 5)	7.34%	8.56%	10.82%	10.24%	11.51%	9.92%	9.73%	3.33%	2.99%	2.50%	2.94%
Pinyin_familyname(len $\geq$ 5)	1.35%	1.64%	2.34%	2.24%	2.47%	1.88%	1.99%	0.05%	0.07%	0.07%	0.06%
Pinyin_fullname(len $\geq$ 5)	**8.39%**	**9.87%**	**12.91%**	**11.81%**	**13.14%**	**11.29%**	**11.24%**	4.79%	4.17%	3.35%	**4.10%**
Pinyin_name_any(len $\geq$ 5)	8.56%	10.05%	13.31%	12.11%	13.46%	11.53%	11.50%	4.80%	4.18%	3.36%	4.11%
Pinyin_place(len $\geq$ 5)	1.24%	1.27%	1.64%	1.58%	2.12%	1.48%	1.55%	0.20%	0.18%	0.16%	0.18%
PW_with_a_5^+-letter_substring	**18.51%**	**19.99%**	**26.95%**	**19.38%**	**28.03%**	**21.70%**	**22.42%**	**71.69%**	**75.93%**	**68.66%**	**72.09%**
Date_YYYY	14.38%	12.82%	12.45%	10.06%	16.91%	14.33%	13.49%	4.34%	4.30%	2.77%	3.80%
Date_YYYYMMDD	6.06%	5.42%	3.93%	3.94%	8.78%	6.17%	**5.72%**	0.10%	0.05%	0.09%	0.08%
Date_MMDD	24.99%	19.97%	17.08%	16.46%	24.45%	22.59%	20.92%	7.53%	4.46%	3.59%	5.20%
Date_YYMMDD	**21.29%**	**15.89%**	**12.70%**	**13.09%**	**20.67%**	**18.28%**	**16.99%**	3.24%	1.23%	1.55%	2.01%
Date_any_above	**36.61%**	**30.39%**	**26.66%**	**27.07%**	**35.30%**	**33.58%**	**31.60%**	11.33%	8.77%	6.45%	8.85%
PW_with_a_digit	89.49%	88.42%	88.52%	90.76%	87.10%	89.26%	88.93%	54.04%	64.74%	46.14%	54.97%
PW_with_a_4^+-digit_substring	81.64%	76.98%	71.90%	78.76%	78.38%	80.60%	78.04%	24.72%	21.85%	19.33%	21.97%
PW_with_a_6^+-digit_substring	**75.59%**	**68.32%**	**61.16%**	**70.02%**	**69.87%**	**73.10%**	**69.68%**	17.77%	8.48%	11.28%	12.51%
PW_with_a_8^+-digit_substring	28.04%	27.56%	26.53%	26.37%	49.73%	31.03%	31.54%	6.88%	2.50%	3.73%	4.37%
Mobile_Phone_Number(11-digit)	**2.90%**	**1.76%**	**2.63%**	**3.97%**	**3.75%**	**2.44%**	***2.91%***	0.07%	0.01%	0.02%	0.03%
PW_with_a_11^+-digit_substring	**4.71%**	**2.09%**	**3.39%**	**5.08%**	**7.57%**	**3.35%**	**4.36%**	0.75%	0.17%	0.18%	0.37%

*Each percentage (%) is counted by the rule of "left-most longest" match and taken by dividing the corresponding password dataset size.

(11.24%) include a 5^+-letter Pinyin full name. There is also 4.10% of English passwords that contain a 5^+-letter full Pinyin name. A reasonable explanation is that many Chinese users have created accounts in these English sites. For instance, the popular Chinese Pinyin 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.

Equally interestingly, we find that, on average, 16.99% of Chinese users insert a six-digit date into their passwords. Further considering that users love to include self information into passwords [9, 56], such dates are likely to be users' *birthdays*. Besides, about 30.89% of Chinese speaking users use a 4^+-digit date to create passwords, which is 3.59 times higher than that of English speaking users (i.e. 8.61%). Also, there are 13.49% of Chinese users inserting a four-digit year into their passwords, which is 3.55 times higher than that of English speaking users (3.80%, which is comparable to the results in [14]). 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 I love you"). As discussed later, we have devised reasonable ways to address this issue. In all, dates play a vital role in passwords of Chinese users.

We mainly pay attention to length-4, 6 and 8 digits in passwords, because: 1) Length-4 and 6 are the most widely used lengths of PINs in the West and Asia; and 2) 6&8 are the two most frequent password lengths (see Fig. 2). It is interesting to see that 2.91% of Chinese users are likely to use their 11-digit mobile numbers as passwords, making up 39.59% of all passwords with an 11^+-digit substring. On 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 of 23.48%($= \frac{2.91\%}{12.39\%}$) 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 unavoidable ambiguities when determining whether a text/digit sequence belongs to a specific dictionary, and an improper resolution of these ambiguities would lead to an overestimation or underestimation. Here we take "YYMMDD" for illustration. For example, both 111111 and 520521 fall into "YYMMDD" and are highly popular. However, 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., 111111 stands for "Nov. 11th, 2011" and 520131 for "Jan 31th, 1952") and completely excluding them from "YYMMDD" would lead to *underestimation* of dates.

Thus, we assume that user birthdays are randomly distributed and assign the expectation of the 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 "YYMMDD", each of which originally has a frequency$> 10E$ and appears in every top-1000 list of the six Chinese datasets. In this way, the ambiguities can be largely resolved. We similarly tackle 16 abnormal items in "MMDD". The detailed info about these abnormal dates can be found in Appendix B. As for the other 19 dictionaries in Table 5, few abnormal items can be identified, and they are processed as usual.

Summary. We have measured nine password datasets in terms of letter distribution, length distribution, frequency distribution and semantic patterns. To our knowledge, most of these fundamental characteristics have at most been mentioned/exampled in the literature (see [26, 30, 34, 36, 53]) but never *systematically* examined. We have identified a number of *similarities* (e.g., frequency distribution and the theme of love) and *differences* (e.g., letter distribution, structural patterns, and semantic patterns) between passwords of these two user groups.

4 Strength of Chinese web passwords

Now we employ two state-of-the-art password attacking algorithms (i.e., PCFG-based [58] and Markov-based [36]) to evaluate the strength of Chinese web passwords. We further investigate whether the characteristics identified in Sec. 3.3 (e.g., dates and Pinyin names) can be practically exploited to facilitate password guessing.

Necessity of pairing passwords by service type. There are a number of confounding factors that impact password security, among which language, service type, and password policy are the three most important ones [29, 53, 56]. As shown in [36, 53], except for CSDN that imposes a length 8^+ policy, all our datasets (Table 1) reflect no explicit policy requirements. It has recently been revealed that users often rationally choose robust passwords for accounts perceived to be important [46], while knowingly choose weak passwords for unimportant accounts [17]. Since accounts of the same service would generally have the same level of value for users, we divide datasets into three pairs according to their types of services (i.e., Tianya vs. Rockyou, Dodonew vs. Yahoo, and CSDN vs. Phpbb) for fairer strength comparison, as opposed to existing works [7, 26, 34] that do not take into account the site service type. We emphasize that it is less reasonable if one compares Dodonew passwords (from an e-commerce site) with Phpbb passwords (from a low-value programmer forum): Even if Dodonew passwords are stronger than Phpbb passwords, one can not conclude that Chinese passwords are more secure than English ones, because there is a potential that Dodonew passwords will be weaker than Yahoo e-commerce passwords.

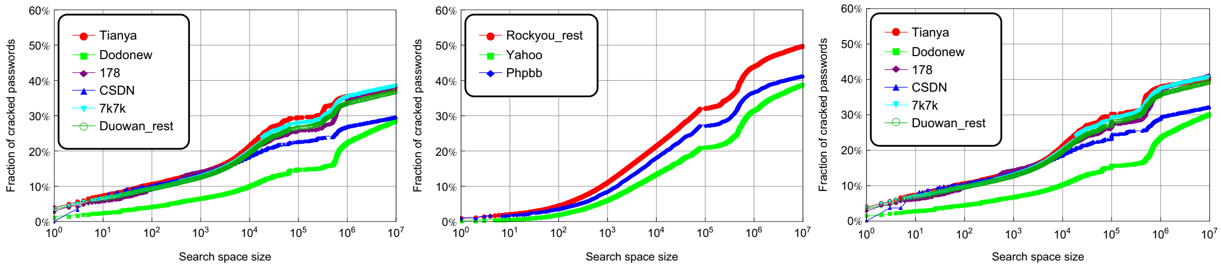
4.1 PCFG-based attacks

The PCFG-based model [58] is one of the state-of-the-art cracking models. Firstly, it divides all the passwords in a training set into segments of similar character sequences and obtains the corresponding base structures and their associated probabilities of occurrence. For example, "`wanglei@123`" is divided into the L segment "wanglei", S segment "@" and D segment "123", resulting in a base structure $\mathsf{L}_7\mathsf{S}_1\mathsf{D}_3$. The probability of $\mathsf{L}_7\mathsf{S}_1\mathsf{D}_3$ is $\frac{\text{\#of } \mathrm{L}_7\mathrm{S}_1\mathrm{D}_3}{\text{\#of base structures}}$. Such information is used to generate the probabilistic context-free grammar.

Then, one can derive password guesses in decreasing order of probability. The probability of each guess is the product of the probabilities of the productions used in its derivation. For instance, the probability of "`liwei@123`" is computed as $P(\text{"}\texttt{liwei@123}\text{"}) = P(\mathsf{L}_5\mathsf{S}_1\mathsf{D}_3) \cdot P(\mathsf{L}_5 \to \text{liwei}) \cdot P(\mathsf{S}_1 \to \texttt{@}) \cdot P(\mathsf{D}_3 \to 123)$. In Weir et al.'s original proposal [58], the probabilities for D and S segments are learned from the training set by counting, yet L segments are handled either by learning from the training set or by using an external input dictionary. Ma et al. [36] revealed that PCFG-based attacks with L segments directly learned from the training set generally perform better than using an external input dictionary. Thus, we prefer to instantiate the PCFG L segments of password guesses by directly learning from the training set.

We divide the nine datasets into two groups by language. 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 is that: Duowan has the least inversion number with the dataset "All Chinese PWs" (see Sec. 3.3) and is likely to best represent general Chinese web passwords. Similarly, for the English test sets, we select 1M passwords from Rockyou as the training set. Since we have only used part of Duowan and Rockyou, their remaining passwords and the other 7 datasets are used as the test sets. The attacking results on the Chinese group and English group are depicted in Fig. 4(a) and Fig. 4(b), respectively.

Bifacial-security. 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. Phpbb). 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 will be even wider when the guess number further increases. This reveals a reversal principle, i.e., the bifacial-security nature of Chinese passwords: they are more vulnerable to online guessing attacks (i.e., when the guess number allowed is small) than English passwords; But out of the remaining Chinese passwords, they are more secure against offline guessing. This reconciles two drastically conflicting claims (see Sec. 1.1) made about the strength of Chinese passwords. This bifacial-security is highly due to the bifacial-density nature of digit-based passwords: *Top* digit-based passwords are



(a) PCFG-based attacks on six Chinese datasets (training set: 1M Rockyou passwords)

(b) PCFG-based attacks on three English datasets (training set: 1M Rockyou passwords)

(c) Improved PCFG-based attacks on Chinese datasets (training set: 1M Rockyou passwords)

Figure 4: General and our improved PCFG attacks on different groups of datasets. Our algorithm gains tangible advantages.

more converging (see Table 3), while digits *in general* are more random (and diverging) than letters.

A weakness in PCFG. We observe that, the original PCFG algorithm [36, 58] inherently gives extremely low probabilities to password guesses (e.g., “1q2w3e4r” and “1a2b3c4d”) that are of a monotonically long base structure (e.g., $D_1L_1D_1L_1D_1$ $L_1D_1L_1$, or $(D_1L_1)_4$ for short). For example, $P(\text{“1q2w3e4r”}) = P((D_1L_1)_4) \cdot P(D_1 \to 1) \cdot P(L_1 \to q) \cdot P(D_1 \to 2) \cdot P(L_1 \to w) \cdot P(D_1 \to 3) \cdot P(L_1 \to e) \cdot P(D_1 \to 4) \cdot P(L_1 \to r)$ can hardly be larger than 10^{-9}, for it is a multiplication of *nine* probabilities. Thus, some guesses (e.g., “1q2w3e4r” and “a12b34c56”) will never appear in the top-10^7 guess list generated by the original PCFG algorithm, even if they are popular (e.g, “1q2w3e4r” appears in the top-200 list of every dataset). The essential reason is that the PCFG 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 $(D_1L_1)_4$ of password “1q2w3e4r” are evidently interrelated with each other (i.e., D_4: 1234 and L_4: qwer).

Our solution. To address this problem, we specially tackle a few password structures that are long but simple alternations of short segments by treating them as short structures. For instance, $(D_1L_1)_4$ is converted to D_4L_4, and $(D_1L_2)_3$ to D_3L_6. In this way, the probability of “1q2w3e4r” now is computed as $P(\text{“1q2w3e4r”}) = P((D_1L_1)_4) \cdot P((D_1L_1)_4 \to D_4L_4) \cdot P(D_4 \to 1234) \cdot P(L_4 \to \text{qwer})$. *Our approach is language-agnostic and constitutes a* general *amendment to the state-of-the -art PCFG-based algorithm in [36].*

To further exploit the characteristics of Chinese passwords, we insert the “Pinyin_name_any” dictionary and the six-digit date dictionary (see Sec. 3.3) into the original PCFG L-segment and D-segment dictionaries, respectively. Details about this insertion process and our improved algorithm for password-guess generation are shown in Algorithm 1. The resulting changes to the original PCFG grammars are given in Table 6.

Algorithm 1: Our improved PCFG-based attack

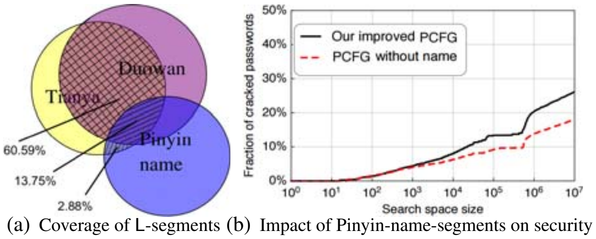
```
Input: A training set $\mathcal{S}$; A name list $nameList$; A date list
       $dateList$; A parameter $k$ indicating the desired size of
       the PW guess list that will be generated (e.g., $k = 10^7$)
Output: A PW guess list $L$ with the top-$k$ items
Training (lastly tackle monotonically long PWs:)
    for $password \in \mathcal{S}$ do
        for $segment \in splitToSegments(password)$ do
            $segmentSet.insert(segment)$
        $baseStructure \leftarrow getBaseStructure(password)$
        if $monotonicallyLong(baseStructure)$ then
            $transformStructureSet.insert(baseStructure)$
            $baseStructure \leftarrow$
              $convertToshort(baseStructure)$
        $baseStructureSet.insert(baseStructure)$
        $trainingSet.insert(password)$
Append name and date lists to the learned segment list:
    for $name \in nameList$ do
        $correctedCount =$
          $totalOverlapNameInSegmentSet *$
          $nameList.getCount(name)/totalOverlapNameInNameList$
        if $name \notin segmentSet$ and $correctedCount \geq 1$ then
            $segmentSet.insert(name, correctedCount)$
    for $date \in dateList$ do
        if $date \notin segementSet$ then
            $segementSet.insert(date)$
Produce $k$ guesses: As with [36] and the details are omitted.
```

Table 6: Changes caused to the original PCFG grammars

Training set	Base structures	L segments	D segments	S segments
Duowan_1M	8905+0	155693+24416	465157+20341	865+0
Duowan_All	20961+0	559017+98654	1824404+9744	2417+0

Fig. 4(c) illustrates that, when the guess number allowed is small (e.g., 10^3), our improved attack exhibits little improvement; As the guess number grows, the improvement increases. For example, at 10^5 guesses, there is 0.09%∼0.85% improvement in success rate; at 10^6 guesses, this figure is 1.32∼4.32%; at 10^7 guesses, this figure reaches 1.70%∼4.29%. This indicates that the vulnerable behaviors of using monotonically long passwords, Pinyin names and birthdays help an attacker reduce her search space, and this issue is more serious when large guesses are allowed.

Comparison. Li et al. [34] reported that using 2M Dodonew passwords as the training set and at 10^{10} guesses, their best success rates $(=\frac{\text{\# of successfully cracked PWs}}{\text{the size of test set}})$ is about 17.30%. However, against the same Chinese test sets, our improved attack can achieve much higher success rates (29.41%∼39.47%) at only 10^7 guesses.



(a) Coverage of L-segments (b) Impact of Pinyin-name-segments on security

Figure 5: Coverage and security impacts of Pinyin-name-segments in the test set Tianya with L-segments involved (Duowan is the training set, Pinyin_name is an extra input dictionary in our improved PCFG attack).

This means that we can crack 70% to 128% more passwords than Li et al.'s best record. Our attacks are better because: 1) Our training-set (i.e., Duowan) is more effective than [34], for we find Duowan represents Chinese password distributions better (see Table 2) than Dodonew as used in [34]; 2) We optimize PCFG not only through adding semantic dictionaries as [34] but also through transforming monotonically long base structures.

The role of Names. 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 Fig. 4(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 attack against Tianya (trained on Duowan) as an example. During training, we have added 98K name segments (see Table 6) into the L-segment dictionary.

Fig. 5(a) demonstrates that these 98K name segments *only* cover 2.88% of the total L segments of the test set Tianya. 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 Tianya. This suggests that Duowan can *well* cover the name segments in the test set Tianya, and thus the addition of some extra names would have limited impacts. This observation also holds for the other eight test sets. The detailed results are summarized in Table 7, 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 7 shows that no matter x=Duowan_1M or Duowan: 1) CoL(x) is at least 11.12 times (= 65.64%/ 5.90%) larger than CoL(Pinyin_name)-CoL(x); 2) CoL(Pinyin_name)∩CoL(x) is at least 1.92 times (=11.35%/ 5.90%) larger than CoL(Pinyin_ name)-CoL(x). This suggests that adding extra 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 web passwords and pose a serious vulnerability. Actually, this *does* suggest that when the training set is selected properly, the name segments in passwords can be well guessed. Still, when there is no proper training set available, our improved attack would demonstrate its advantages (see Fig. 5(b)). Though our improved PCFG algorithm might not be optimal, its cracking results represent a new benchmark that any future algorithm should aim to decisively clear.

Limitations. We mainly investigate the impacts of names on password cracking, and similar observations and implications are likely to hold for dates (but with no confirmation). We leave it as future work. In addition, as our focus is the overall security of Chinese passwords (and its comparison with English counterparts), we only show the overall effectiveness of our improved PCFG attack. It is also interesting to see to what extent the improved PCFG structure and the usage of Duowan would respectively have impacts on the cracking effectiveness, but it is independent of the presented work.

Table 8: Five Markov-based attacking scenarios

Attacking scenario	Smoothing	Normalization	Markov order
#1	Laplace	End-symbol	3/4/5
#2	Laplace	Distribution	3/4/5
#3	Good-Turing	End-symbol	3/4/5
#4	Good-Turing	Distribution	3/4/5
#5	Backoff	End-symbol	Backoff

4.2 Markov-based attacks

To show the robustness of our findings about password security, we further conduct Markov-based attacks.

4.2.1 Markov-based experimental setups

To make our experiments as reproducible as possible, we now detail the setups. As recommended in [36], we consider two smoothing techniques (i.e., Laplace Smoothing and Good-Turing Smoothing) to deal with the data sparsity problem and two normalization techniques (i.e., distribution-based and end-symbol-based) to deal with the unbalanced length distribution problem of passwords. This brings four attacking scenarios in Table 8. 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 that another scenario (i.e., backoff with end-symbol normalization) performs "slightly better" than the above 4 scenarios, yet it is "approximately 11 times slower, both for guess generation and for probability estimation" [36]. We also investigate this scenario and observe similar results. Thus, attackers, who particularly care about the cost-effectiveness [4], are highly unlikely to exploit this scenario.

Particularly, there is a challenge to be addressed when implementing the Good-Turing (GT) smoothing technique. To our knowledge, we for the first time explicate how to combine GT and simple GT in Markov-based

Table 7: Coverage of letter (CoL) segments in corresponding test sets ("PY" stands for Pinyin).

Test set	CoL (PY_name)	CoL (Duowan1M)	CoL(PY_name)∩ CoL(Duowan1M)	CoL(PY_name)− CoL(Duowan1M)	CoL(Duowan1M) −CoL(PY_name)	CoL (Duowan)	CoL(PY_name) ∩CoL(Duowan)	CoL(PY_name) −CoL(Duowan)	CoL(Duowan)− CoL(PY_name)
Tianya	16.63%	67.53%	11.82%	**4.81%**	55.71%	74.34%	13.75%	**2.88%**	60.59%
7k7k	16.70%	71.60%	12.35%	**4.35%**	59.25%	79.84%	14.49%	**2.20%**	65.35%
Dodonew	15.76%	75.79%	11.79%	**3.97%**	63.99%	81.19%	13.47%	**2.29%**	67.72%
178	20.30%	79.15%	15.42%	**4.88%**	63.73%	83.98%	17.49%	**2.81%**	66.49%
CSDN	17.26%	65.64%	11.35%	**5.90%**	54.28%	72.70%	13.43%	**3.83%**	59.27%
Duowan	18.06%	80.05%	14.38%	**3.68%**	65.67%	100.00%	18.06%	**0.00%**	81.94%
Duowan_rest	18.07%	75.03%	13.46%	**4.61%**	61.57%	100.00%	18.07%	**0.00%**	81.93%

attacks (see details in Appendix C). As with PCFG-based attacks, in our implementation we use a max-heap 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^{-10}) for the probability of guesses generated, then sorting all the guesses, and finally selecting the top k ones. In this way, we can 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 δ to the count of each substring and we set δ=0.01 as suggested in [36].

4.2.2 Markov-based experimental results

The experiment results for these five scenarios are quite similar. Here we mainly show the cracking results of Scenario #1 in Fig. 6, while the experiment results for Scenarios #2∼#5 are omitted due to space constraints.

We can see that, for both Chinese and English test sets: (1) At large guesses (i.e., $>2*10^6$), order-4 markov-chain evidently performs better than the other two orders, while at small guesses (i.e., $<10^6$) the larger the order, the better the performance will be; (2) There is little 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 normalization always performs better than the distribution-based approach, while at small guesses its advantages will be more obvious. Such observations have not been reported in previous major studies [15, 36]. This suggests that: 1) At large guesses, the attacks with order-4, Laplace Smoothing and end-symbol normalization (see Figs. 6(b) and 6(e)) perform best; and 2) At small guesses, the attacks preferring order-5, Laplace Smoothing and end-symbol normalization (see Figs. 6(a) and 6(d)) perform best.

Results show that *the bifacial-security nature found in our PCFG attacks (see Sec. 5.1) also applies in all the Markov attacks*. For example, in order-4 markov-chain-based experiments (see Fig.6(b) and Fig.6(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 allowed is over 10^4, Chinese web passwords are generally *stronger* than their English counterparts. For example, at 10^6 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.

Table 9: Bifacial-security nature of Chinese passwords.[†]

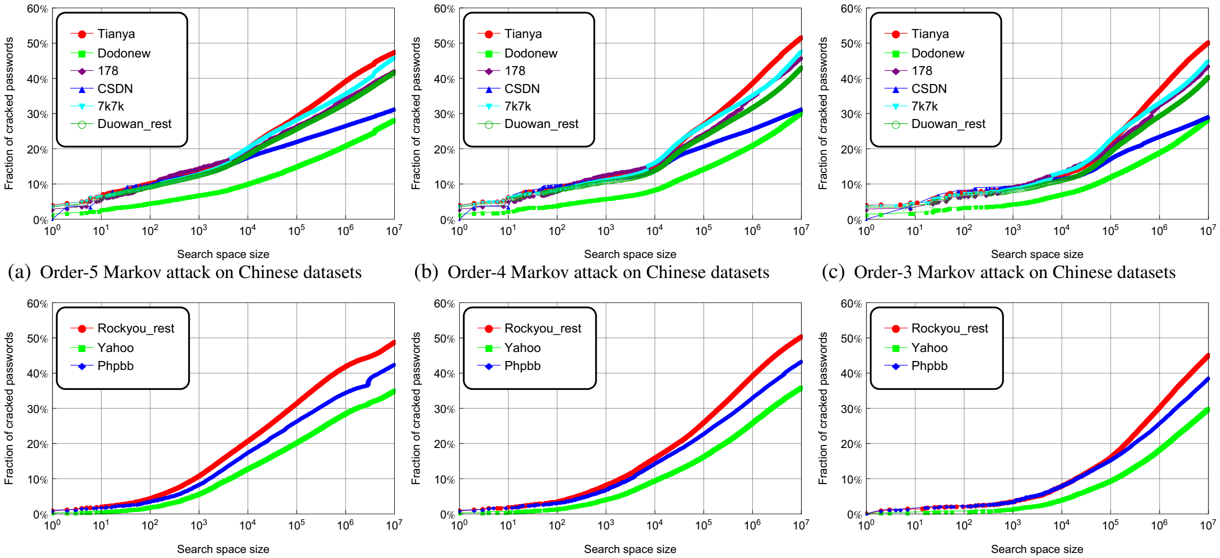
Attacking scenario		Online guessing			Offline guessing			
Algorithm*	Test set	10^1	10^2	10^3	10^4	10^5	10^6	10^7
PCFG	Dodonew	**0.027**	**0.044**	**0.068**	0.103	0.150	0.225	0.288
	Yahoo	0.008	0.022	0.063	**0.136**	**0.212**	**0.316**	**0.390**
	Tianya	**0.073**	**0.105**	**0.138**	0.213	0.295	0.355	0.376
	Rockyou_rest	0.020	0.044	0.110	**0.214**	**0.320**	**0.438**	**0.497**
	CSDN	**0.070**	**0.105**	**0.136**	0.189	0.229	0.272	0.300
	Phpbb	0.021	0.038	0.087	**0.183**	**0.274**	**0.369**	**0.415**
Markov	Dodonew	**0.024**	**0.040**	**0.060**	0.085	0.145	0.212	0.305
	Yahoo	0.007	0.016	0.043	**0.097**	**0.165**	**0.261**	**0.361**
	Tianya	**0.062**	**0.087**	**0.118**	0.154	0.269	0.386	0.516
	Rockyou_rest	0.018	0.035	0.081	**0.159**	**0.259**	**0.392**	**0.503**
	CSDN	**0.037**	**0.098**	**0.116**	0.144	0.211	0.260	0.316
	Phpbb	0.019	0.034	0.071	**0.146**	**0.230**	**0.333**	**0.436**

[†] A value in bold **green** (e.g., the leftmost 0.027) means that: it is a success-rate under a given guess number (resp. 10^1) against a Chinese dataset (resp. Dodonew) and is *greater* than that of its English counterpart (resp. Yahoo). A value in bold **blue** is on the contrary: it is a guessing success-rate against a English dataset and *greater* than that of its Chinese counterpart.

*For both PCFG- and Markov-based attacks, the training set is Duowan_1M for each Chinese test set and Rockyou_1M for English test sets. Here the Markov setups are from Scenario#1 in Table 8. Other Markov scenarios show the same trends.

As summarized in Table 9, for both PCFG and Markov attacks, the cracking success-rates against Chinese passwords are *always higher* than those of English passwords when the guess number is below 10^4, while this trend is reversed when the guess number is above 10^4. Here we mainly use order-4 Markov attacks (see Figs. 6(b) and 6(e)) as an example, and the other Markov setup scenarios all show the same trends.

Summary. Both PCFG- and Markov-based cracking results reveal the bifacial-security nature of Chinese passwords: They are more prone to online guessing as compared to English passwords; But out of the remaining Chinese passwords, they are more secure against offline guessing. This reconciles the conflicting claims made in [7, 26, 34]. Alarmingly high cracking rates (40%∼50%) highlight the urgency of developing defense-in-depth countermeasures (e.g., cracking-resistant honeywords [31] and password-hardening services [33]) to alleviate the situation. We provide a large-scale empirical evidence for the hypothesis raised by the HCI community [17, 46]: users rationally choose stronger passwords for accounts with higher value.



(a) Order-5 Markov attack on Chinese datasets
(b) Order-4 Markov attack on Chinese datasets
(c) Order-3 Markov attack on Chinese datasets
(d) Order-5 Markov attack on English datasets
(e) Order-4 Markov attack on English datasets
(f) Order-3 Markov attack on English datasets

Figure 6: Markov-chain-based attacks on different groups of datasets (scenario #1: *Laplace Smoothing* and *End-Symbol Normalization*). Attacks (a)∼(c) use 1 million Duowan passwords as the training set, while attacks (d)∼(f) use 1 million Rockyou passwords as the training set. The reversal principle also holds. The other four scenarios #2∼#5 show similar cracking results.

5 Some implications

We now elaborate on lessons learned and key takeaways.

5.1 For password creation policies

Interestingly, 2.18% of the passwords in CSDN are of length $len \leq 7$, 97.82% are of length 8-20, no password is of length $len \geq 21$. This means that short passwords (i.e., $len \leq 7$) in the other eight sites are 14∼25 times higher than CSDN. This also suggests that CSDN has changed its password policy at least once before the data breach (i.e., Dec. 2011), but whether the strict policy (i.e., $8 \leq len \leq 20$) is enforced before or later than the weaker policy (i.e., no length requirement) is unknown.[3] Still, what's certain is that most CSDN passwords are generated under the strict policy $8 \leq len \leq 20$. In contrast, no apparent policy can be inferred from the Dodonew data, i.e., neither minimum length (see Fig. 2) nor charset requirement (see Table 3 and Table 2 of [53]).[4] However, Figs. 4 and 6 indicate that, given any guess number below 10^7, passwords from CSDN are significantly weaker than passwords from Dodonew. A plausible reason is that Dodonew provides e-commerce services and users perceive it as more important. As a result, users "rationally" [17, 46] choose more complex passwords for it. As for CSDN, since it is only a technology forum, users knowingly choose weaker passwords for it.

In 2012, Bonneau [7] cast doubt on the hypothesis that users rationally select more secure passwords to protect their more important accounts. In 2013, Egelman et al. [17] initiated a field study involving 51 students and confirmed this hypothesis. In 2018, Stobert and Biddle [46] surveyed three groups of English speaking users (i.e., 27 non-experts, 15 experts and 345 MTurk participants), and their results also corroborated this hypothesis. Fortunately, our work provides a large-scale empirical evidence (i.e., on the basis of 6.43M CSDN passwords and 16.26M Dodonew passwords) that confirms this hypothesis.

We also note that though the overall security of Dodonew passwords is higher than that of passwords from the five other Chinese sites, many seemingly complex yet popular passwords (e.g., `5201314`, `321654a` and `love4ever`) dwelling in Dodonew also appear in other less sensitive sites. This can be understood: 1) "Users never see other users' passwords" [47] (and are unaware of how similar their passwords are with other users, and thus they may inadvertently choose popular passwords; 2) Users tend to reuse the same password across multiple sites [27, 42, 56]. What's more, users generally "show a lack of situation awareness" [46] and fail to recognize different categories of accounts [41], and most of them reuse (84% [27]) or simply modify a password from an important site for a non-important site.

Further considering the great password burden already placed on users [8] and the "bounded rationality" [27] and "finite-effort" [20] of users, we outline the need for

[3] We note that CSDN enforced the policy $6 \leq len \leq 20$ (and no charset requirement) at Jan. 2015 [55], and currently it requires passwords to be $11 \leq len \leq 20$ and consist at least a letter and a digit.

[4] This situation even held at Aug. 2017 (and April 2019): the length-7 letter string "dodonew" is allowed as the default password, see `https://www.5636.com/netbar/money/15886.html`.

HCI research to explore nudges that appropriately frame the importance of accounts and study their impacts on password creation. When designing password creation policies, instead of merely insisting on stringent rules, administrators can employ such nudges to help users gain *more accurate perceptions of the importance* of the accounts to be protected and improve their *ability to recognize different categories* of accounts. Both would help enhance user internal impetus and facilitate users to responsibly allocate passwords (i.e., selecting one candidate from their limited pool of passwords memorized [41,46]).

In addition, the finding of "bifacial-security nature" suggests that Chinese passwords are more vulnerable to online guessing attacks. This is because top popular Chinese passwords are more concentrated (see Table 3). Thus, a special blacklist that includes a moderate number of most common Chinese passwords (e.g., 10K∼20K as suggested in [61]) would be very helpful for Chinese sites to resist against online guessing. Such a blacklist can be learned from various leaked Chinese datasets (see a concrete list at `http://t.cn/RG88tvF` as built according to [56]). Any password falling into this list would be deemed weak. However, it is well known that if some popular passwords (e.g., `woaini1314`) are banned, new popular ones (e.g., `w0aini1314`) will arise. These new popular passwords may be out of static blacklists and subtle to detect. Hence, password creation policies alone (e.g., length and blacklist rules [25, 55]) are inadequate for preventing such weak passwords. An in-depth defense approach is needed: whenever possible, in addition to password creation policies, password strength meters (e.g., fuzzyPSM [54] and Zxcvbn [59]) can be further employed by security-critical services to detect and prevent weak passwords.

5.2 For password strength meters

Leading password strength meters (PSMs) employ the guess number needed for a password-cracking algorithm (e.g. PCFG) to break that password as an indicator of password strength [24]. In Sec. 1, we have exemplified that the PSMs of four popular services are highly inconsistent in assessing the security of (weak) Chinese passwords. Failing to provide accurate/coherent feedback on user password choices would have negative effects such as user confusion, frustration and distrust [48,60]. Thus, Carnavalet and Mannan [12] suggested that PSMs "can simplify challenges by limiting their primary goal only to detect *weak* passwords, instead of trying to distinguish a good, very good, or great password."

It follows that an essential step of a PSM would be to identify the characteristics of weak passwords. From our findings in Section 3.3 and Section 4.1, it is evident that for passwords of Chinese users, the incorporation of long Pinyin words or full/family names is an important evidence/weight for a "weak" decision. Other signs of weak Chinese passwords are the incorporation of birthdates and simple patterns like repetition, palindrome and keyboard. As a caveat, even if signs of weak passwords are found, one cannot simply deem such passwords as weak and reject them as is done in many high-profile sites (e.g., Microsoft Azure [18]) and by the "substring blacklist" approach recommended in [44]. Instead, such undesirable/insecure signs should be weighted (see some promising attempts in [54,59]).

The superiority of our improved PCFG-based attacks over Li et al.'s [34] (see Sec. 4.1) is partly attributed to the proper selection of Duowan (instead of Dodonew as in [34]) as the training set. This indicates that, for a PSM to be accurate, its training set should be representative of the password base of the target site. The distance of letter distributions (see Table 2) would be an effective metric. In addition, the universal "bifacial-security nature" revealed in Sec. 4 implies that, the language factor is more impactful than service type. We also find that CSDN passwords are weaker than Dodonew passwords (see Figs. 4 and 6), but CSDN imposes a stricter policy than Dodonew, and this suggests that the service-type factor might be more impactful than password policy.

Thus, when measuring the letter distributions is infeasible, these con-founding factors underlying a password distribution can be considered for training-set selection: 1) In the order of language, service, and password policy; and 2) The closer the training set to the target password, the better. This suggests that there is no single training set that can fit all PSMs. Thus, PSMs that are originally designed for English speaking users and also do *not* employ a training set (e.g., NIST entropy [10], RNN-PSM [38] and Zxcvbn [59]) cannot be readily applied to Chinese users. This also explains why such PSMs are generally less accurate than those using a training set (e.g., fuzzyPSM [54]) as observed in [24].

5.3 For password cracking

Password cracking algorithms are not only necessary tools for security administrators to measure password strength, but also they can be used to facilitate information forensics (e.g., for law enforcement agencies to recover encrypted data of criminal suspects). Three main lessons for password cracking can be learned from our above results. Firstly, our findings in Sec. 3.3 show that Chinese passwords have a vastly different letter distribution, structure and semantic patterns as compared to English passwords, and thus when targeting a Chinese password, it is crucial for cracking algorithms to be trained on datasets from Chinese sites. Such sites should also have the same password creation policy and the same (or a similar) service type as the target site.

Secondly, for PCFG-based attacks, when the training set is sufficiently large (e.g., over 1M as ours), besides the D and S-segments, it is better to also directly learn the L-segments of guesses from the training set. This can be well established by the fact that, given the same guess numbers and against the same test sets, our PCFG-based attacks can obtain much higher success rates (see Sec. 4.1) than those of the PCFG-based attacks in [34,58] where external dictionaries are used to instantiate the L-segments. This practice has been recommended by Ma et al. [36], but they did not specify when to apply it. Further, one may include some external semantic dictionaries to instantiate the L and D-segments as we do.

Thirdly, as compared to Markov-based attacks, PCFG-based ones are simpler to implement (31% less computation and 70% less memory cost), and they perform equally well, or even better, when the guess number is small (e.g. 10^3, see Figs. 4 and 6). For large guess numbers, order-4 Markov attacks are the best choices. As far as we know, these observations have not been elucidated in previous major studies [15, 36]. Note that, we have only shown the Markov-based cracking results when the guess number is below 10^7. There is potential that order-3 Markov-based attacks will outperform order-4 and 5 ones at larger guess numbers (e.g., 10^{14}).

6 Conclusion

In this paper, we performed a large-scale empirical analysis of 73.1 million real-world Chinese web passwords. In our empirical analysis, we systematically explored several fundamental password properties (e.g., the distance between passwords and languages, and various semantic patterns) and uncovered the bifacial-security nature of Chinese passwords: They are more prone to online guessing than English passwords; But out of the remaining Chinese passwords, they are stronger against offline guessing. This reconciles two conflicting claims in [7, 26, 34]. We hope this work will help both security administrators and individual Chinese users to more informedly secure their password accounts.

Acknowledgment

The authors are grateful to Mary Ellen Zurko for shepherding our paper. We thank Haibo Cheng, Qianchen Gu, and anonymous referees for invaluable help and comments. Ping Wang is the corresponding author. This research was supported by the National Natural Science Foundation of China under Grants No. 61802006 and No. 61572379, and by the National Key Research and Development Plan under Grant No.2017YFB1200700.

References

[1] *China now has 802 million internet users*, July 2018, http://n0.sinaimg.cn/tech/c0a99b19/20180820/CNNIC42.pdf.

[2] M. AlSabah, G. Oligeri, and R. Riley, "Your culture is in your password: An analysis of a demographically-diverse password dataset," *Comput. Secur.*, vol. 77, pp. 427–441, 2018.

[3] J. Blocki, A. Datta, and J. Bonneau, "Differentially private password frequency lists," in *Proc. NDSS 2016*, pp. 1–15.

[4] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," in *Proc. IEEE S&P 2018*, pp. 35–53.

[5] J. Bonneau, "Guessing human-chosen secrets," Ph.D. dissertation, University of Cambridge, 2012.

[6] J. Bonneau, C. Herley, P. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE S&P 2012*, pp. 553–567.

[7] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE S&P 2012*, pp. 538–552.

[8] J. Bonneau, C. Herley, P. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Comm. ACM*, vol. 58, no. 7, pp. 78–87, 2015.

[9] A. S. Brown, E. Bracken, and S. Zoccoli, "Generating and remembering passwords," *Applied Cogn. Psych.*, vol. 18, no. 6, pp. 641–651, 2004.

[10] W. Burr, D. Dodson, R. Perlner, S. Gupta, and E. Nabbus, "NIST SP800-63-2: Electronic authentication guideline," National Institute of Standards and Technology, Reston, VA, Tech. Rep., 2013.

[11] R. A. Butler, *List of the Most Common Names in the U.S.*, Jan. 2018, http://names.mongabay.com/most_common_surnames.htm.

[12] X. Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *Proc. NDSS 2014*.

[13] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from markov models," in *Proc. NDSS 2012*.

[14] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," in *Proc. NDSS 2014*, pp. 1–15.

[15] M. Dell'Amico and M. Filippone, "Monte carlo strength evaluation: Fast and reliable password checking," in *Proc. ACM CCS 2015*, pp. 158–169.

[16] M. Dürmuth, D. Freeman, and B. Biggio, "Who are you? A statistical approach to measuring user authenticity," in *Proc. NDSS 2016*, pp. 1–15.

[17] S. Egelman, A. Sotirakopoulos, K. Beznosov, and C. Herley, "Does my password go up to eleven?: the impact of password meters on password selection," in *Proc. ACM CHI 2013*, pp. 2379–2388.

[18] *Eliminate bad passwords in your organization*, July 2018, https://docs.microsoft.com/bs-latn-ba/azure/active-directory/authentication/concept-password-ban-bad.

[19] D. Florêncio and C. Herley, "A large-scale study of web password habits," in *Proc. WWW 2007*, pp. 657–666.

[20] D. Florêncio, C. Herley, and P. C. Van Oorschot, "Password portfolios and the finite-effort user: Sustainably managing large numbers of accounts," in *Proc. USENIX SEC 2014*, pp. 575–590.

[21] S. Furnell and R. Esmael, "Evaluating the effect of guidance and feedback upon password compliance," *Comput. Fraud Secur.*, vol. 2017, no. 1, pp. 5–10, 2017.

[22] W. Gale and G. Sampson, "Good-turing smoothing without tears," *J. Quanti. Linguistics*, vol. 2, no. 3, pp. 217–237, 1995.

[23] J. Goldman, *Chinese Hackers Publish 20 Million Hotel Reservations*, Dec. 2013, http://www.esecurityplanet.com/hackers/chinese-hackers-publish-20-million-hotel-reservations.html.

[24] M. Golla and M. Dürmuth, "On the accuracy of password strength meters," in *Proc. ACM CCS 2018*, pp. 1567–1582.

[25] P. A. Grassi, E. M. Newton, R. A. Perlner, and et al., "NIST 800-63B digital identity guidelines: Authentication and lifecycle management," McLean, VA, Tech. Rep., June 2017.

[26] W. Han, Z. Li, L. Yuan, and W. Xu, "Regional patterns and vulnerability analysis of chinese web passwords," *IEEE Trans. Inform. Foren. Secur.*, vol. 11, no. 2, pp. 258–272, 2016.

[27] A. Hanamsagar, S. S. Woo, C. Kanich, and J. Mirkovic, "Leveraging semantic transformation to investigate password habits and their causes," in *Proc. ACM CHI 2018*, pp. 1–10.

[28] J. Huang, H. Jin, F. Wang, and B. Chen, "Research on keyboard layout for chinese pinyin ime," *J. Chin. Inf. Process.*, vol. 24, no. 6, pp. 108–113, 2010.

[29] M. Jakobsson and M. Dhiman, "The benefits of understanding passwords," in *Proc. HotSec 2012*, pp. 1–6.

[30] S. Ji, S. Yang, X. Hu, and et al., "Zero-sum password cracking game," *IEEE Trans. Depend. Secur. Comput.*, vol. 14, no. 5, pp. 550–564, 2017.

[31] A. Juels and R. L. Rivest, "Honeywords: Making password-cracking detectable," in *Proc. ACM CCS 2013*, pp. 145–160.

[32] D. V. Klein, "Foiling the cracker: A survey of, and improvements to, password security," in *Proc. of USENIX SEC 1990*, pp. 5–14.

[33] R. W. Lai, C. Egger, M. Reinert, S. S. Chow, M. Maffei, and D. Schröder, "Simple password-hardened encryption services," in *Proc. Usenix SEC 2018*, pp. 1405–1421.

[34] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis on chinese web passwords," in *Proc. USENIX SEC 2014*.

[35] B. Lu, X. Zhang, Z. Ling, Y. Zhang, and Z. Lin, "A measurement study of authentication rate-limiting mechanisms of modern websites," in *Proc. ACSAC 2018*, pp. 89–100.

[36] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *IEEE S&P 2014*, 2014, pp. 689–704.

[37] M. L. Mazurek, S. Komanduri, T. Vidas, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur, "Measuring password guessability for an entire university," in *Proc. ACM CCS 2013*, pp. 173–186.

[38] W. Melicher, B. Ur, S. Segreti, and et al., "Fast, lean and accurate: Modeling password guessability using neural networks," in *Proc. USENIX SEC 2016*, pp. 1–17.

[39] R. Morris and K. Thompson, "Password security: A case history," *Comm. ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[40] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proc. ACM CCS 2005*, pp. 364–372.

[41] R. Nithyanand and R. Johnson, "The password allocation problem: strategies for reusing passwords effectively," in *Proc. ACM WPES 2013*, pp. 255–260.

[42] S. Pearman, J. Thomas, P. E. Naeini, and et al., "Let's go in for a closer look: Observing passwords in their natural habitat," in *Proc. ACM CCS 2017*, pp. 295–310.

[43] B. L. Riddle, M. S. Miron, and J. A. Semo, "Passwords in use in a university timesharing environment," *Comput. Secur.*, vol. 8, no. 7, pp. 569–579, 1989.

[44] R. Shay, S. Komanduri, A. L. Durity, and et al., "Designing password policies for strength and usability," *ACM Trans. Inform. Syst. Secur.*, vol. 18, no. 4, pp. 1–34, 2016.

[45] *Sogou Internet thesaurus*, Sogou Labs, April 17 2018, http://www.sogou.com/labs/dl/w.html.

[46] E. Stobert and R. Biddle, "The password life cycle," *ACM Trans. Priv. Secur.*, vol. 21, no. 3, pp. 1–32, 2018.

[47] B. Ur, J. Bees, S. M. Segreti, L. Bauer, N. Christin, L. F. Cranor, and A. Deepak, "Do users' perceptions of password security match reality?" in *Proc. ACM CHI 2016*, pp. 1–10.

[48] B. Ur, P. G. Kelley, S. Komanduri, and et al., "How does your password measure up? the effect of strength meters on password creation," in *Proc. USENIX SEC 2012*, pp. 65–80.

[49] L. Vaas, https://nakedsecurity.sophos.com/2016/08/16/people-like-using-passwords-way-more-than-biometrics/.

[50] A. Vance, *If Your Password Is 123456, Just Make It HackMe*, Jan. 2010, https://www.nytimes.com/2010/01/21/technology/21password.html.

[51] R. Veras, J. Thorpe, and C. Collins, "Visualizing semantics in passwords: The role of dates," in *Proc. ACM VizSec 2012*, pp. 88–95.

[52] C. Wang, S. T. Jan, H. Hu, D. Bossart, and G. Wang, "The next domino to fall: Empirical analysis of user passwords across online services," in *Proc. CODASPY 2018*, pp. 196–203.

[53] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inform. Foren. Secur.*, vol. 12, no. 11, pp. 2776–2791, 2017.

[54] D. Wang, D. He, H. Cheng, and P. Wang, "fuzzyPSM: A new password strength meter using fuzzy probabilistic context-free grammars," in *Proc. IEEE/IFIP DSN 2016*, pp. 595–606.

[55] D. Wang and P. Wang, "The emperor's new password creation policies," in *Proc. ESORICS 2015*, pp. 456–477.

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

[57] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proc. ACM CCS 2010*, pp. 162–175.

[58] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *Proc. IEEE S&P 2009*, pp. 391–405.

[59] D. Wheeler, "zxcvbn: Low-budget password strength estimation," in *Proc. USENIX SEC 2016*, pp. 157–173.

[60] *Why is Gbt3fC79ZmMEFUFJ a weak password?*, Jan. 2019, https://security.stackexchange.com/questions/201210/why-is-gbt3fc79zmmefufj-a-weak-password.

[61] R. Williams, *The UX of a blacklist*, Mar. 2018, https://news.ycombinator.com/item?id=16434266.

[62] J. Yan, A. F. Blackwell, R. J. Anderson, and A. Grant, "Password memorability and security: Empirical results." *IEEE Secur. Priv.*, vol. 2, no. 5, pp. 25–31, 2004.

[63] C. Zuo, W. Wang, R. Wang, and Z. Lin, "Automatic forgery of cryptographically consistent messages to identify security vulnerabilities in mobile services," in *Proc. NDSS 2016*.

APPENDIX

A Justification for our cleaning approach

Contaminated datasets. Interestingly, we observe 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" was originally in the top-10 most popular list of both datasets. We manually scrutinized 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 posit that someone probably has copied these joint accounts from one dataset to the other.

Our cleaning approach. Now, a natural question arises: *From which dataset have these joint accounts been copied?* We conclude 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` naturally include the site name "`tianya`" into their passwords for convenience. The following second reason is quite subtle yet convincing. In the original Tianya dataset, the joint accounts are of the form {user name, email address, password}, while in the original 7k7k dataset such joint accounts are 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 $64822/2 < 48871$. Thus, it is more plausible for users 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 first copy all the accounts (i.e., 64822/2) using "`111222tianya`" as the password from 7k7k to Tianya and then 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}. 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 cleaning is summarized in Table 1.

Previous studies. In 2014, Li et al. [34] has also exploited the datasets Tianya and 7k7k. However, contrary to us, they think that the 3.91M joint accounts are copied from 7k7k to Tianya. Their main 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 did not explain what the "various compositions" are. Their explanation also does not answer the critical question: why are there so many 7k7k users using "`111222tianya`" as their passwords? We posit they had removed 3.91*2 million joint accounts from 7k7k but the not 3.91 million ones from Tianya In addition, they did not observe the extremely *abnormal* fact that all the passwords in 7k7k occur even times. Such contaminated data would lead to inaccurate results. For example, Li et al. [34] reported that there are 32.41% of passwords in 7k7k containing dates in "YYYMMDD", yet the actual value is 6 times lower: 5.42%.

We have reported this issue to the authors of [34], they responded to us and acknowledged this flaw in their journal version [26]. Unfortunately, Han et al. [26] do not clean the datasets in the journal version in the manner that we outlined.

B Detailed information about our 22 semantic dictionaries

In order to make our work as reproducible as possible and to facilitate the community, we now detail how to construct our 22 semantic-based dictionaries. All dictionaries are built with natural lengths. The length≥5 requirement in the upper-part of Table 5 is set *conservatively* for ensuring accuracy only when we perform matching. Actually, we also performed measurements for length≥3 and length≥4, and got higher figures (percentages) but less accuracy. Thus, we omit them.

The first dictionary "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 names with over 0.001% frequency in the US population during the 1990 census, according to the US Census Bureau [11]. "English_firstname" contains 5,494 most common first names (including 1,219 male and 4,275 female names) in US [11]. The dictionary "English_fullname" is a cartesian product of "English _firstname" and "English_lastname", consisting of 1.04 million most common English full names.

To get a Chinese full name dictionary, we employ the 20 million hotel reservations dataset [23] 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 "Pinyin_fullname" and "Pinyin_familyname", 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 [45] and also originally in Chinese. 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 pre-defined 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 `https://shouji.supfree.net/`.

As for the birthday dictionaries, we use date patterns to match digit strings that might be birthdays. For example, "YYYYMMDD" 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). Similarly, we build the date dictionaries "YYYY", "MMDD" and "YYMMDD". Note that, "PW with a l^+-letter substring" means 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".

Though we use the "left-most longest" rule to minimize ambiguities when matching, there are some unavoidable ambiguities when determining whether a text/digit sequence belongs to a semantic dictionary. An improper resolution would lead to an overestimation or underestimation. For instance, `111111` falls into "YYMMDD" and is highly popular, yet it is more likely that users choose it simply because it is easily memorable repetition numbers. To tackle this issue, we manually identify 17 abnormal dates in "YYMMDD", each of which originally has a frequency$> 10E$ and appears in every top-1000 list of the six Chinese datasets: `111111`, `520131`, `111222`, `121212`, `520520`, `110110`, `231231`, `101010`, `110119`, `321123`, `010203`, `110120`, `010101`, `520530`, `000111`, `000123`, `080808`. Similarly, we identify 16 abnormal items in "MMDD": `1111`, `1122`, `1231`, `1212`, `1112`, `1222`, `1010`, `0101`, `1223`, `1123`, `0123`, `1020`, `1230`, `0102`, `0520`, `1110`. Few abnormal items can be identified in the other 19 dictionaries (Table 5), and they are processed as usual.

C A subtlety about Good-Turing smoothing in Markov-based cracking

In 2014, Ma et al. [36] introduced the Good-Turing (GT) smoothing into password cracking, yet little attention has been paid to the unsoundness of GT for popular password segments. We illustrate the following subtlety.

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 "$c_1 c_2 \cdots c_l$" in a Markov model of order n is denoted by

$$P(\text{“}c_1 \cdots c_{l-1} c_l\text{”}) = \prod_{i=1}^{l} P(\text{“}c_i | c_{i-n} c_{i-(n-1)} \cdots c_{i-1}\text{”}), \quad (1)$$

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

$$P(\text{“}c_i | c_{i-n} \cdots c_{i-1}\text{”}) = \frac{S(count(c_{i-n} \cdots c_{i-1} c_i))}{\sum_{c \in \Sigma} S(count(c_{i-n} \cdots c_{i-1} c))}, \quad (2)$$

where the alphabet Σ includes 95 printable ASCII characters on the keyboard (plus one special end-symbol c_E denoting the end of a password), and $S(\cdot)$ is defined as:

$$S(f) = (f+1)\frac{N_{f+1}}{N_f}. \quad (3)$$

This kind of smoothing works well when f is small, but 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 Rockyou and occurs $f = 490{,}044$ times. Since there is no 5-character string that occurs 490,045 times, N_{490045} will be zero, implying the basic GT estimator will set P("`12345`")=0. A similar problem regarding the smoothing of password frequencies is identified in [5].

There have been various improvements suggested in linguistics to tackle this problem, among which is the "simple Good-Turing smoothing" [22]. This improvement (denoted by SGT) is famous for its simplicity and accuracy. SGT takes two steps of smoothing. Firstly, SGT performs a smoothing operation for N_f:

$$SN(f) = \begin{cases} N(1) & \text{if } f = 1 \\ \dfrac{2N(f)}{f^+ - f^-} & \text{if } 1 < f < \max(f) \\ \dfrac{2N(f)}{f - f^-} & \text{if } f = \max(f) \end{cases} \quad (4)$$

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 resulting from regression. Finally, SGT conducts a second smoothing by replacing the raw count N_f from Eq.3 with $Z(f)$:

$$S(f) = \begin{cases} (f+1)\dfrac{N_{f+1}}{N_f} & \text{if } 0 \le f < f_0 \\ (f+1)\dfrac{Z(f+1)}{Z(f)} & \text{if } f_0 \le f \end{cases} \quad (5)$$

where $t(f) = |(f+1) \cdot \frac{N_{f+1}}{N_f} - (f+1) \cdot \frac{Z(f+1)}{Z(f)}|$ and $f_0 = \min\left\{ f \in \mathbb{Z} \,\middle|\, N_f > 0, t(f) > 1.65\sqrt{(f+1)^2 \frac{N_{f+1}}{N_f^2}(1 + \frac{N_{f+1}}{N_f})} \right\}$.

To the best of our knowledge, we for the first time well explicate how to combine the two smoothing techniques (i.e., GT and SGT) in Markov-based password cracking.