Zipf's Law in Passwords

Ding Wang[1], Gaopeng Jian[2], Haibo Cheng[2], Qianchen Gu[1], Chen Zhu[1], Ping Wang[1]

[1] School of EECS, Peking University, Beijing 100871, China
[2] School of Mathematical Sciences, Peking University, Beijing 100871, China
{wangdingg,demscimath,chenghaibo,qcgu,czhu,pwang}@pku.edu.cn

Abstract. Despite more than thirty years of research efforts, textual passwords are still enveloped in mysterious veils. In this work, we make a substantial step forward in understanding the distributions of passwords and measuring the strength of password datasets by using a statistical approach. We first show that Zipf's law perfectly exists in real-life passwords by conducting linear regressions on a corpus of 56 million passwords. As one specific application of this observation, we propose the number of unique passwords used in regression and the slope of the regression line together as a metric for assessing the strength of password datasets, and prove it in a mathematically rigorous manner. Furthermore, extensive experiments (including optimal attacks, simulated optimal attacks and state-of-the-art cracking sessions) are performed to demonstrate the practical effectiveness of our metric. To the best of knowledge, our new metric is the first one that is both easy to approximate and accurate to facilitate comparisons, providing a useful tool for the system administrators to gain a precise grasp of the strength of their password datasets and to adjust the password policies more reasonably.

Keywords: User authentication, Password cracking, Statistics

1 Introduction

User authentication is the first line of defense for information systems to safeguard resources and services from unauthorized access. Even though much has been reported about their pitfalls, textual passwords are still the dominant mechanism of Internet authentication, protecting hundreds of millions of accounts on Internet-scale websites. Recently, there have been countless attempts in proposing alternative authentication schemes (e.g., two-factor authentication [40], graphical passwords [11]) to dislodge passwords, yet passwords are more widely used and firmly entrenched than ever. As passwords offer many advantages not always matched by other alternative schemes [6] and moreover, the transition costs of replacing them can not be effectively quantified [18], they are likely to persist in the foreseeable future.

Despite its ubiquity, password authentication is accompanied by the dilemma of generating passwords which are both challenging for powerful attackers to crack and easy for common users to remember. Truly random password is difficult for users to memorize, while user-chosen password may be highly predictable [48]. In practice, users tend to choose passwords that are related to their daily lives, which means these passwords are drawn from a rather small dictionary [5] and thus are vulnerable to guessing attacks.

To mitigate this notorious "security-usability" dilemma, there have proposed various password creation policies (e.g., random generation [48], rule-based [4, 42], entropy-based [8, 44] and cracking-based [10, 19, 39]) to force newly created passwords to adhere to some requirements (rules) and to achieve an acceptable strength. The diversity of password strength meters and rules brings about an enormous variety of requirements between different websites, resulting in highly conflicting strength outcomes for the same password [9], e.g., the password `password$1` is deemed "Very Weak" by Dropbox, "Fair" by Google and "Very Strong" by Yahoo!

The above conflicting password strength outcomes (for more concrete examples, see [9]) are a direct result of the inconsistent password strength meters employed among different websites, which may be further explained by the diverse interests of each website. It is generally believed that stricter policies might make passwords harder to crack, but the side effect is that users may feel harder to create and to remember passwords and thus usability is reduced [43]. Inglesant and Sasse [20] also reported that, inappropriate password policies in a specific context of use can increase both mental and cognitive workload on users and impact negatively on their productivity and, ultimately, that of the organisation and the circumvention of such policies.

As a result, different type of websites typically may have quite different favors. For commercial retailers like Amazon, portals like Yahoo! and advertising supported sites like Facebook, usability is very important because every login event is a revenue opportunity. Anything that interferes with user experience affects the business directly. So they tend to have less restrictive password policies. On the other hand, it's critical to prevent attackers

from illicitly accessing valuable resources for cloud storage sites (e.g., Dropbox) that maintain sensitive documents and university sites that manage course grades. So they may require that user-selected passwords are subject to more complex constraints (e.g., inclusion of mixed-case, digits and special characters, and rejection of popular passwords like `pa$$word123`).

1.1 Motivations

Usually, the services provided by a website and its users may vary as time goes on, which may lead to huge changes in the user password dataset after some period of time (e.g., one year) even though the password policy stays the same. In this situation, the system administrators of a website shall quantify the strength of the whole passwords and may need to adjust the password policy. Either failing to notice the changes in the password dataset or conducting improper countermeasures may rise great but subtle security and usability problems as illustrated above. So a proper assessment of the strength of password dataset is essential, without which the system administrator is unable to determine the following critical question: How shall the password policy be adjusted? To put it another way, shall the password policy be enhanced to improve security, kept unchanged or even relaxed a bit to get usability in return? The essence of designing a password creation policy is to accurately gauge the strength of individual passwords, while the heart of appropriately adjusting the password creation policies is to accurately assess the strength of given password datasets. Note that, in this work we presume that a state-of-the-art password creation policy (e.g., [10, 19]) has already been adopted by the authentication system, and its adjustment mainly involves changing some rules and the password strength threshold.

Surprisingly, as far as we know, existing literature does not provide a satisfactory answer to the above question of how to accurately measure the strength of a given password dataset. The settlement of this question will naturally entail the settlement of the following question: How to precisely characterize a given password dataset? And again, little progress has been made on this question. The two most commonly used approaches to assess the strength of a given password dataset are estimating its information entropy (e.g., [8]) and empirically analyzing its "guessability" (e.g., [23, 45]). The former, however, is not based on empirical data and has been shown inaccurate [45], and the latter, which largely depends on the choices of the cracking algorithms and input dictionaries [14, 27], has too many uncertainties to accurately characterize the strength of a given dataset.

Very recently, Bonneau [5] introduced a novel statistical-based metric parameterized by an attacker's desired success rate α and it is called α-guesswork, yet its effectiveness has not been testified by empirical results using different datasets. In addition, α-guesswork is unsuitable for comparing the strength of two datasets in many cases, for the comparison results may vary with α and thus no deterministic conclusion can be made. This failure is mainly due to the lack of an appropriate characterization of the distribution of passwords.

To the best of our knowledge, Malone and Maher's work [29] may be the most relevant to what we will discuss in the current paper. They investigated the distribution of passwords, however, contrary to what we will show in this work, it was concluded that "while a Zipf distribution does not fully describe our data, it provides a reasonable model, particularly of the long tail of password choices." According to this conclusion, a Zipf distribution cannot fully characterize the passwords, then what will work?

1.2 Our contributions

In this work, we bring the evaluation of real-life password datasets onto a sound scientific footing by adapting statistical techniques and provide definite answers to the above-mentioned two interesting (and important) questions: (1) *How to precisely characterize a given password dataset?* and (2) *How to accurately measure the strength of a given password dataset?*

Our first contribution is to adopt techniques from computational statistics to demonstrate that Zipf's law perfectly exists in real-life passwords, inspired by the applicability of Zipf's law to describe surprisingly diverse natural and social phenomena like Linux software distribution [28] as well as US firm sizes [2]. We prune these least frequent passwords and rank the frequency of each remaining unique password (either in plain-text or hashed form) in decreasing order and investigate the mathematical relationships between the frequency and the rank by using linear regression. Extensive experiments on a massive corpus of eight password datasets different in terms of size, application domain and user localization, show that our model is able to accurately characterize the distribution of real-life passwords, particularly of the front head of passwords, but not "of the long tail of password choices" that is reported in [29]. This provides a satisfactory answer to the first question above.

Our second contribution is that we propose a novel metric to facilitate the system administrators to have a concise grasp of the strength of their password datasets in a mathematically rigorous manner, and enable them to

compare the results of their own website at different time points or compare with that of other websites which are meaningful for reference (at least the datasets revealed in this paper can be used as good counterpoints). Based on these comparison and assessment results, the system administrators now can make their choice more wisely than before–whether to continue using the current password policy, enhance it, or even relax it a bit to improve usability? This suggests the settlement of the above first question.

Our last contribution is to show the effectiveness of our metric for measuring the strength of password datasets by simulating optimal cracking attacks on eight real-life password datasets. We take a step further to employ the state-of-the-art cracking algorithm (i.e., probabilistic context-free grammars (PCFG) [23, 46]) to approximate optimal password cracking attack. Of independent interest may be our observation that PCFG-based cracking results (i.e., success rates) are much lower as compared to optimal results, which implies that the state-of-the-art cracking algorithm is far from an optimal one and there leaves much room for future improvement. What's more, Bonneau's α-guesswork [5] is further developed.

Roadmap. In Section 2, we survey related works. We show Zipf's law exists in passwords in Section 3. Our password dataset strength metric is presented, proved, and empirically established in Section 4. Section 5 conclues the paper.

2 Related Work

In this section, we discuss some related works on password creation policies and password cracking techniques.

2.1 Password creation policies

In 1990, Klein proposed the concept of proactive password checker, which enables users to create passwords and checks, a priori, whether the new password is "safe" [24]. The criteria can be divided into two types. One type is the exact rules for what constitutes an acceptable password, such as minimum length requirements and character type requirements. The other type is using a reject function based on estimated password strength. An example of this is a blacklist of "weak" passwords that are not allowed. Although the author calls the technique "proactive password checking", it's indeed the same as password creation policies we know today, so in the following we use the two terms interchangeably.

Since Klein's seminal work, there have been proposed a number of proactive password checkers that aim to reduce the time and space of matching newly-created passwords with a blacklist of "weak" passwords, such as Opus [42] and BApasswd [13]. There have also been attempts to design tuneable rules on a per-site basis to shape password creation, among which is the influential NIST Electronic Authentication Guideline SP-800-63 [8]. However, by modeling the success rates of current password cracking techniques against real-life user passwords under different rules, Weir et al. [45] showed that merely rule-based policies perform poorly for ensuring a desirable level of security.

In 2012, on the basis of Weir et al.'s work [45], Houshmand et al. [19] proposed a novel policy that first analyzes whether a user selected password is weak or strong according to empirical PCFG cracking results, and then modifies the password slightly if it is weak to create a strengthened password. This policy facilitates the measurement of the strength of individual passwords more accurately and in addition, it can be adjusted more flexibly than previous policies due to the fact that an adjustment of the policy only involves tuning the threshold with continuous ranges.

Perhaps the most relevant policy related to our strength metric for assessing password datasets (see Sec.4) is suggested by Schechter et al. [39]. Their intriguing idea is to use a popularity oracle to replace traditional password creation policies, and thus passwords with high popularity are rejected. This policy is particularly effective at thwarting statistical-based guessing attacks against Internet-scale authentication systems that have millions of user accounts. If this policy is in place, our proposed metric would be largely unnecessary. However, how to prevent an attacker from using their oracle to guess passwords is an open question. Moreover, as this approach rejects passwords that occur at a frequency exceeding a threshold r (e.g., $r = \frac{1}{10^6}$ as exampled in their work [39]), which would frequently frustrate users from using their previous passwords that are (sufficiently) popular. For instance, about 34.89% user in Tianya.cn use passwords that are more frequent than $r = \frac{1}{10^6}$, which indicates that more than one third of the users will be annoyed to use a new password. Whether this would greatly reduce usability has not been evaluated thoroughly, and it is likely to pose as a serious problem.

2.2 Password cracking

Password-based systems are prone to various attacks, including on-line guessing, offline guessing, keylogging and social engineering [26]. Here we only consider the on-line and offline guessing attacks, because other attacks where

the attacker obtains the password by non-cryptographic ways, are unrelated to password strength or password dataset strength and therefore outside the scope of this work. While online guessing attacks can be well thwarted by no-cryptographic techniques, such as locking an account after a threshold number of failed logins or using more flexible lockout strategies [1], offline guessing attacks are offline performed and not subject to the defender's security measures, and thus the attacker can make as many guesses as possible given enough time and computational power.

Consequently, it is essential for password-based systems to properly evaluate their resilience to offline guessing attacks. In the literature, this is generally done by comparing the search space size (i.e., number of guesses) against the percentage of hashed passwords that would be broken by an offline attack. This measure only depends on the attacking technique and the way users choose their passwords, it is neither related to the particular nature of the authentication system nor affected by the attacker capabilities. The nature of the system and attacker capabilities will instead define the cost that the attacker has to pay for each single guess [14]. For example, system countermeasures against offline guessing attacks, such as salting to defeat pre-computation techniques (e.g., Rainbow tables [35]) or key strengthening techniques [16] to make guessing attacks more costly, only constitute a key parameter when evaluating the resilience of a password system to offline attacks. By combining this cost with a measure of the search space, it becomes possible to attain a concise cost-benefit analysis for offline attacks. This kind of measure is also followed by our work.

Password search space essentially depends on how the users choose their passwords. It is a well known fact that users tend to choose mnemonic passwords [36]. However, users rarely use unmodified elements of such lists, for instance, because password creation policies prevent this practice, and instead users modify the words in such a way that they can still recall them easily. For example, the popular password `password$` is generated by adding one symbol to the easily guessable string `password`.

To model this password generation practice, researchers utilize various heuristic mangling rules to produce variants of words from an input dictionary like [36], and this sort of techniques have emerged as early as 1979 in Morris and Thompson's analysis of 3,000 passwords [32]. This initial work has been followed by independent studies by Klein [24] and Spafford [41]. Later on, some dedicated software tools like John the Ripper [15] emerged. Subsequent studies (e.g., [25,47]) have often utilized these software tools to perform dictionary attacks as a secondary goal.

It was not until very recently that password cracking began to deviate from art to science. In 2005, Narayanan and Shmatikov [33] proposed a new cracking algorithm that uses Markov chain instead of ad hoc mangling rules to model user password creation patterns. Their algorithm generates passwords that are phonetically similar to words and is tested on a dataset of 142 hashed passwords and 96 (67.6%) passwords were successfully broken. Yet, their algorithm is not a standard dictionary-based attack, for it can only produce linguistically likely passwords. Moreover, the test dataset is too limited to convincingly show its effectiveness.

In 2009, on the basis of probabilistic context-free grammars, Weir et al. [46] proposed a novel technique for automatically deriving word-mangling rules, and they further employed large real-life datasets to test its effectiveness. In this technique, a password is considered as a combination of alphabet symbols (denoted by L), digits (D) and special characters (S). For instance, password `pa$$word123` is denoted by $L_2S_2L_4D_3$. Then, a set of word-mangling rules is obtained from a training set of clear-text passwords. Coupled with another input dictionary, these rules can be further used to generate password guesses in decreasing probability order, where the probability of each guess is the product of the probabilities of the mangling rules used in its derivation. To simulate the optimal attack, this algorithm generates password guesses in decreasing probability order, and is able to crack 28% to 129% more passwords than John the Ripper [15]. In 2010, Zhang et al. [49] found Weir et al.'s algorithm is the most effective one among all techniques (including Markov model [33], John the Ripper [15] and Rainbow [35]) they used, which has also been confirmed by [5, 10]. Hence, in this work we also use Weir et al.'s algorithm to crack the collected datastets and make comparisons based on the proposed metric.

3 The Zipf's Law in real-life passwords

3.1 Linear regression

In statistics, linear regression is an approach for modeling the relationship between two variables by fitting a linear equation to the observed data. One variable is considered to be an explanatory variable, and the other one is considered to be a dependent variable. Usually, linear regression refers to a model in which the conditional mean of y given the value of x is an affine function of x: $y = a + b \cdot x$, where x is the explanatory variable and y is the dependent variable. The slope of the line is b, and a is the intercept.

The most common method for fitting a regression line is by using least-squares [34]. This method computes the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each

Table 1. Statistical information about eight password datasets

Dataset	Total Passwords	Unique Passwords	[a-z]+	[A-Z]+	[A-Za-z]+	[0-9]+	[a-zA-Z0-9]+	[a-z]+[0-9]+	[a-z]+1	[a-zA-Z]+[0-9]+	[0-9]+[a-zA-Z]+	[0-9]+[a-z]+
Tianya	30,233,633	12,614,676	9.96%	0.18%	10.29%	63.77%	98.05%	14.63%	0.12%	15.64%	4.37%	4.11%
Dodonew	16,231,271	11,236,220	8.79%	0.27%	9.37%	20.49%	82.88%	40.81%	1.39%	42.94%	7.31%	6.95%
CSDN	6,428,287	4,037,610	11.64%	0.47%	12.35%	45.01%	96.31%	26.14%	0.24%	28.45%	6.46%	5.88%
Duowan	4,982,740	3,119,070	10.30%	0.09%	10.52%	52.84%	97.59%	23.97%	0.37%	24.84%	6.04%	5.83%
Myspace	41,545	37,144	7.18%	0.31%	7.66%	0.71%	89.95%	65.66%	18.24%	69.77%	6.02%	5.66%
Singles.org	16,250	12,234	60.20%	1.92%	65.82%	9.58%	99.78%	17.77%	2.73%	19.68%	1.92%	1.77%
Faithwriters	9,709	8,347	54.40%	1.16%	59.04%	6.35%	99.57%	22.82%	4.13%	25.45%	2.73%	2.37%
Hak5	2,987	2,351	18.61%	0.27%	20.39%	5.56%	92.13%	16.57%	2.01%	31.80%	1.44%	1.21%

data point to the line. For example, if a point lies on the fitting line exactly, then its vertical deviation is 0. More specifically, from experiments we collect a bunch of data: $(x_i, y_i), 1 \leq i \leq N$. We expect $y = a + b \cdot x + \varepsilon$, where a, b are constants and ε is the error. If we choose $b = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$ and $a = \bar{y} - b\bar{x}$, where $\bar{x}$ is the arithmetical mean of x_i, similar for $\bar{y}$. Then the sum of the squares of the errors $\sum_{i=1}^{N}(y_i - a - b \cdot x_i)^2$ is minimized. In regression, the coefficient of determination (denoted by R^2 and ranging from 0 to 1) is a statistical measure of how well the regression line approximates the real data points: the closer to 1 the better.

3.2 Description of password datasets

In our work we collect eight large-scale real-life password lists different in terms of application domain, size and user localization, showing that our model is able to accurately characterize the distribution of real-life passwords.

All eight datasets were compromised by hackers or leaked by anonymous insiders, and were subsequently disclosed publicly on the Internet. We realize that while publicly available, these datasets contain private data. Therefore, we treat all passwords as confidential such that using it in our research does not increase the harm to the victims. Furthermore, attackers are likely to use these password sets as training sets or cracking dictionaries, while our use of them are likely to be of practical relevance to security administrators and common users.

The first dataset is the "Myspace" which was originally published in October 2006. Myspace is a famous social networking website in the United States and its passwords were compromised by an attacker who set up a fake Myspace login page and then conducted a standard phishing attack against the users. While several versions of the Myspace dataset exist, owing to the fact that different researchers downloaded the list at different times, we get one version from [7] which contained 41545 plain text passwords.

The following two datasets are the "Singles.org" and the "Faithwriters". They both are composed of people almost exclusively of the Christian faith– www.singles.org is a dating site ostensibly for Christians and www.faithwriters.com an online writing community for Christians. The former was broken into via query string injection and 16250 passwords were leaked, while the latter was compromised by an SQL injection attack which disclosed 9709 passwords.

The fourth dataset is from www.hak5.org and it was compromised by a group called ZF0 (Zero for 0wned) [12]. This dataset is only a small portion of the entire www.hak5.org dataset. Surprisingly, though Hak5 is claimed to be "a cocktail mix of comedy, technolust, hacks, DIY mods, homebrew, forensics, and network security", its dataset is amongst the weakest ones (see Sec.3.3) of all the datasets. In this work, we use this dataset as a counterexample for representatives of real-life password distributions.

The next four datasets, namely Duowan, CSDN, Tianya and Dodonew, are all from Chinese websites. We name these password datasets according to the corresponding websites' domain name (e.g. the "Tianya" dataset is from www.tianya.cn). They are all publicly available on the Internet due to several security breaches that happened in China in December, 2011 [30] and we collected them at that time. CSDN is the largest community website of Chinese programmers, Tianya is an influential Chinese BBS, and Duowan is a popular game forum. All the passwords except part of the "Duowan" dataset are in plain-text. "Duowan" contains both hashed (MD5) and plain-text passwords, and we limit our analysis to the plain-text ones.

Some statistical information about our datasets are summarized in Table 1. In all cases, some users choose the same password, and this is mainly due to the longstanding bad practice of using popular passwords. For example, we find that the top 20 most popular passwords in CSDN dataset account for 11.12% of all the passwords. It is interesting to note that Chinese users are more likely to use only digits to construct their passwords, while English users are more likely to use letters to construct their passwords. A plausible explanation may be that Chinese users, who usually use hieroglyphics and are less familiar with English letters on keyboards. Another interesting observation is that, English users prefer to generate their passwords by adding the digit "1" to a sequence of lower letters.

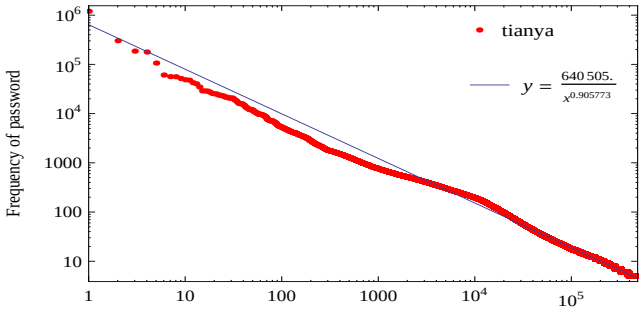


Fig. 1. Zipf's law in Tianya ($R^2 = 0.994$)

3.3 Our findings

Table 2. Linear regression results of password datasets ("PWs" stands for passwords)

Dataset	Totoal PWs	Least frequency	Total PWs in regression	Fraction of PWs in regression	Unique PWs in regression (N)	Absolute value of the slope (s)	Zipf regression line (log$y$)	Coefficient of determination(R^2)
Tianya	30,233,633	5	15,250,838	0.504432861	486,118	0.905773	$5.806523 - 0.905773{*}\log x$	0.994204954
Dodonew	16,231,271	5	3,512,595	0.216409115	187,901	0.753771	$4.618284 - 0.753771{*}\log x$	0.995530686
CSDN	6,428,287	5	1,918,282	0.298412625	57,715	0.894307	$4.886747 - 0.894307{*}\log x$	0.985106832
Duowan	4,982,740	5	1,427,734	0.286535922	51,797	0.841926	$4.666012 - 0.841926{*}\log x$	0.976258449
Myspace	41,545	3	3,363	0.080948369	706	0.459808	$1.722674 - 0.459808{*}\log x$	0.965861431
Singles.org	16,250	3	3,597	0.221353846	658	0.518096	$1.875405 - 0.518096{*}\log x$	0.970277755
Faithwriters	9,709	3	1,211	0.124729632	242	0.486348	$1.583425 - 0.486348{*}\log x$	0.974175889
Hak5	2,987	3	460	0.154000671	76	0.643896	$1.579116 - 0.643896{*}\log x$	0.922662999

Initially, probabilistic context-free grammar (PCFG) is a machine learning technique used in natural language processing (NLP), yet Weir et al. [46] managed to exploit it to automatically build mangling rules. Very recently, NLP techniques have also been shown useful in evaluating the effect of grammar on the vulnerability of long passwords and passphrases by Rao et al. [38].

Inspired by these earlier works, in this study for the first time we attempt to investigate whether the Zipf's law, which resides in natural languages, also exists in passwords. The Zipf's law was first formulated as a rank-frequency relationship to quantify the relative commonness of words in natural languages by Zipf in 1949 [50]. It states that given some corpus of natural language utterances, the frequency of any word is inversely proportional to its rank in the frequency table. More specifically, for a natural language corpus listed in frequency decreasing order, the rank r of a word and its frequency f_r are inversely proportional $f_r = \frac{C}{r}$, where C is a constant depending on the particular corpus. Hence, the most frequent word will occur approximately twice as often as the second most frequent word, three times as often as the third most frequent word, and so on. Recently, Zipf's law has been shown to account remarkably well for the open source Linux software distribution [28], gene expression [17], as well as US firm sizes [2].

Interestingly, by excluding the least popular passwords from the datasets (i.e., passwords with less than three or five counts in this work) and using linear regression, we find that the distribution of real-life passwords obeys a similar law: For a password dataset, the rank r of a password and its frequency f_r follow the equation

$$f_r = \frac{C}{r^s} \tag{1}$$

where C and s are constants depending on the chosen dataset. Zipf's law can be more easily observed by plotting the data on a log-log graph (base 10), with the axes being log(rank order) and log(frequency). In other words, $\log(f_r)$ is linear with $\log(r)$:

$$\log f_r = \log C - s \cdot \log r \tag{2}$$

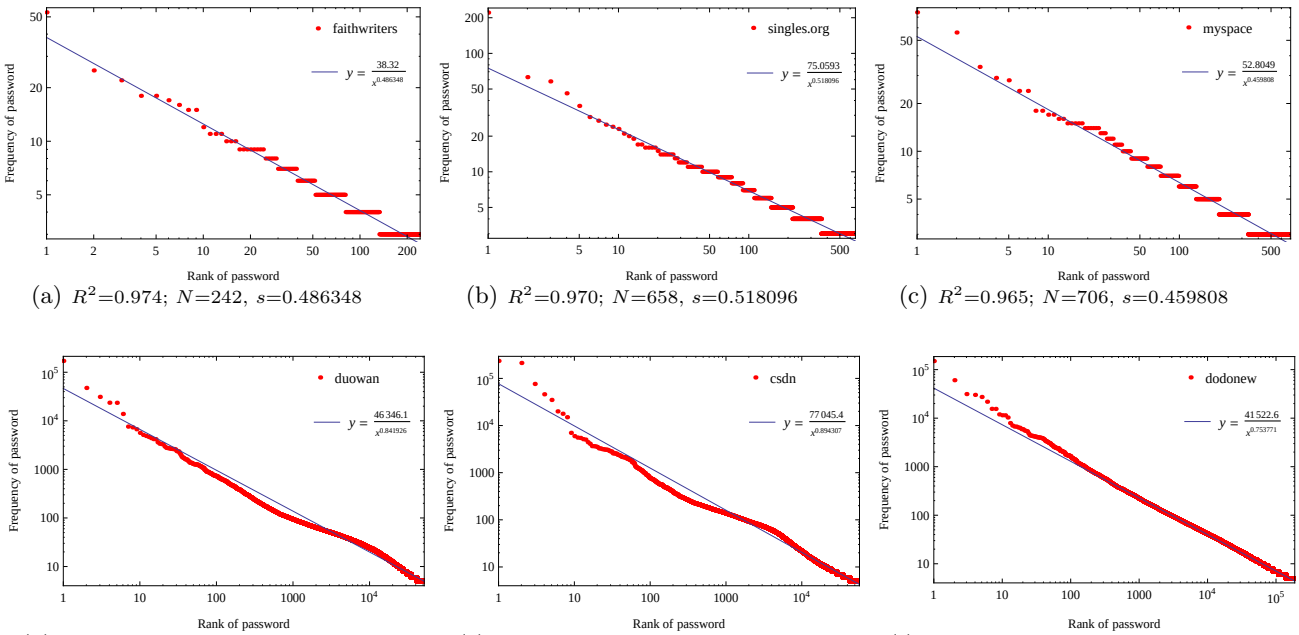
As can be seen from Fig.1, 30 million passwords from the website www.tianya.cn conform to Zipf's law to such extent that the coefficient of determination (denoted by R^2) is 0.994204954, which approximately equals 1. This indicates that the regression line $\log y = 5.806522 - 0.905773{*}\log x$ perfectly fits the data from Tianya. As illustrated in the miniatures in Fig.2, passwords from the other six datasets also invariably adhere to Zipf's law and the regression line well fits the data points from corresponding datasets. Due to space constraints and the

aforementioned imperfect nature of Hak5 dataset, we do not present its related Zipf curve here, though actually its fitting line also has a high coefficient of determination (i.e., $R^2 = 0.923$).

More precisely, as summarized by the "Coefficient of determination" column in Table 2, every linear regression (except for Hak5) is with its R^2 larger than 0.965, which very much approaches to 1 and thus indicates a remarkably sound fitting. As for "Hak5", its R^2 is about 0.923, which is, though acceptable, but not as good as that of other datasets. A plausible reason may be that it only contains less than three thousand passwords and probably can not represent the real distribution of the entire password dataset of www.hak5.org. It also should be noted that, how the datasets leak may have a direct effect on R^2. As confirmed by Table 2, datasets leaked by phing attacks are likely to have a lower R^2 as compared to that of datasets leaked by website breaches, for the former is unlikely to obtain the entire dataset of a website, while the latter, once succeed, all (or at least a complete part of) passwords of the website will be harvested.

Two other critical parameters involved in the regression process are N and s, which stand for the number of unique passwords used in regression and the absolute value of the slope of regression line, respectively. While there is no obvious relationships between N and s, we find there is a close linking between s and the fraction of passwords (or equally, total passwords) used in regression: the larger s is, the larger the latter will be. Once again, dataset Hak5 is an exception and the reasons have been stated earlier.

It is worth mentioning that, as said earlier, we have excluded the passwords with less than five frequencies from the Chinese datasets and with less than three frequencies from the English datasets when performing linear regression, for we conjecture that it is only these popular passwords that will affect (reduce) the strength of a dataset. This conjecture will be established by both rigorous proofs and extensive empirical experiments in the following section.



(a) R^2=0.974; N=242, s=0.486348 (b) R^2=0.970; N=658, s=0.518096 (c) R^2=0.965; N=706, s=0.459808

(d) R^2=0.976; N=51797, s=0.841926 (e) R^2=0.985; N=57715, s=0.894307 (f) R^2=0.996; N=187901, s=0.753771

Fig. 2. Zipf's law in real-life passwords plotted on a log-log scale

4 Strength metric for password dataset

In this section, we pay attention to the question as to how to accurately measure the strength of a given password dataset. As one specific (and natural) application of our observation of the distribution of passwords, an elegant and accurate metric is suggested.

4.1 Our metric

Normally, the attacker, who is clever, would always *attempt* to try the most probable password first and then the second most probable password and so on in decreasing order of probability until success. In the extreme case, if the attacker has also obtained the entire password dataset in plain-text (and thus, the attacker can obtain the right order of the passwords), this attack is called an optimal attack [5, 14]. Accordingly, we can use the cracking result $\lambda^*(n)$ to be the strength metric of a given password dataset:

$$\lambda^*(n) = \frac{1}{sum} \sum_{r=1}^{n} f_r \tag{3}$$

where *sum* is the number of total passwords and n the number of guessing.

In the last section we have shown that the distribution of passwords obeys Zipf's law, i.e., $f_r = \frac{C}{r^s}$. Consequently, $\lambda^*(n)$ is essentially determined by N and s (Note that N is the number of unique passwords, and s is the absolute value of the slope of the fitting line):

$$\lambda^*(n) \approx \lambda(n) = \frac{\sum_{r=1}^{n} \frac{1}{r^s}}{\sum_{r=1}^{N} \frac{1}{r^s}} \tag{4}$$

It should be note that, in Eq.4, $\lambda^*(n)$ is not exactly equal to the value of rightmost hand even though our regression line complies with the actual data very well. We plot $\lambda^*(n)$ as a function of n according to Eq.3 and $\lambda(n)$ as a function of n according to Eq.4, and put these two curves together to see how they agree with each other. In Fig.3, we depict $\lambda^*(n)$ and $\lambda(n)$ for 30 million passwords from the Tianya dataset and obtain an average deviation of 1.32% (i.e., a sound fitting) for the two curves. As explained in Sec.3.3, here we do not illustrate the related picture for Hak5 dataset. As for the other six datasets, see the miniatures in Fig.4.

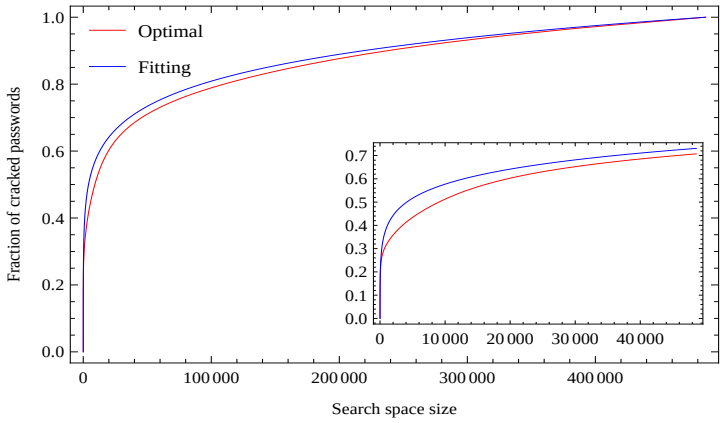


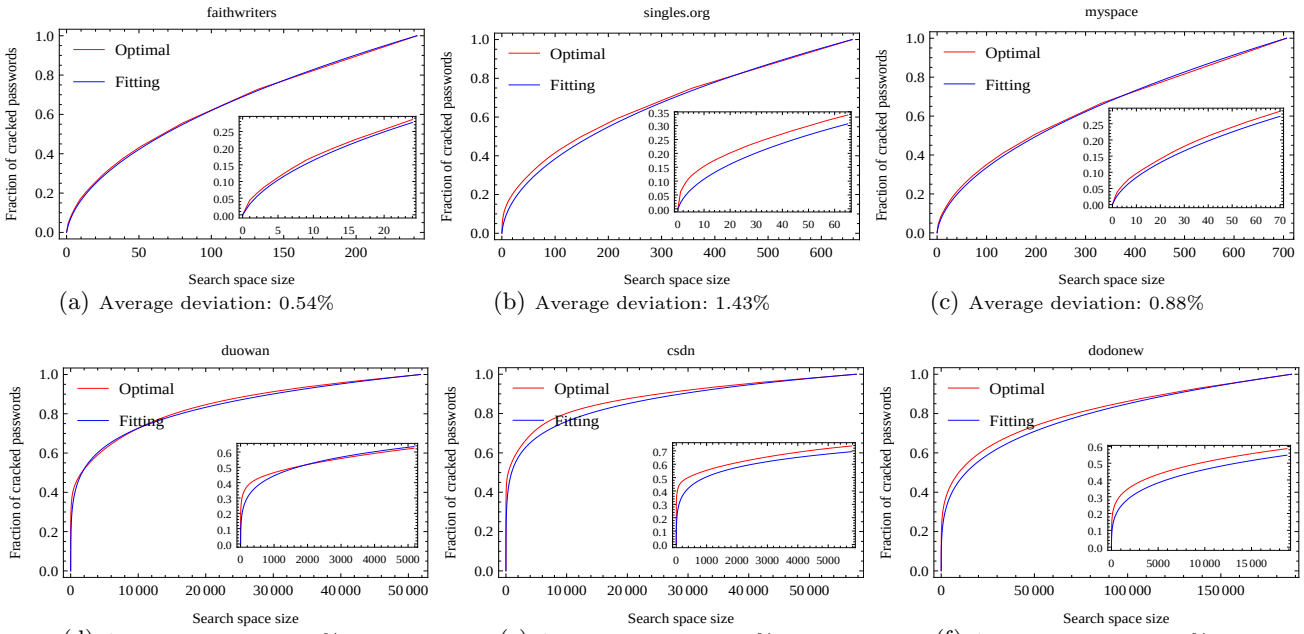
Fig. 3. Consistence of optimal attack with our fitting metric on Tianya (Average deviation is 1.32%)

We can see that, the $\lambda^*(n)$ curve well overlaps with the $\lambda(n)$ curve for each dataset. Specifically, except for Hak5, the average deviations for these datasets are from 0.54% to 1.93%, which shows perfect consistence of $\lambda(n)$ with the optimal attacking results. Note that, the two curves first deviate slightly when n is small and then gradually merge into each other as n increases. This is caused by the variation of the first few high-frequency passwords to the fitting line.

Now that the optimal attack can be well approximated by $\lambda(n)$, it is natural to propose the pair (N_A, s_A) to be the metric for measuring the strength of password dataset A, where N_A is the number of unique passwords used in regression and s_A is the absolute value of the slope of the fitting line. In the following, we propose a theorem and a corollary, and show that our metric not only is able to determine whether the strength of a website's password dataset becomes weak after a period of time, but also can be used to compare the strength of datasets from different websites. This feature is rather appealing, for the confidence of security only comes after comparison– having a comparison with other similar websites, the system administrators now have a clearer picture about what level of strength their datasets can provide. Recent tens of catastrophic leakages of web accounts (see [21] for an incomplete list) provide wonderful materials to facilitate such comparisons.

Theorem 1. *Suppose* $N_A \geq N_B, s_A \leq s_B$. *Then*

$$\lambda_A(n) \leq \lambda_B(n)$$



(a) Average deviation: 0.54% (b) Average deviation: 1.43% (c) Average deviation: 0.88%

(d) Average deviation: 0.86% (e) Average deviation: 1.93% (f) Average deviation: 1.76%

Fig. 4. Consistence of effectiveness between real optimal attack and our fitting metric

where $0 \leq n \leq N_A$ *(if* $n > N_B$*, define* $\lambda_B(n) = 1$*). If either inequalities of the above two conditions are strict, then* $\lambda_A(n) < \lambda_B(n)$*, where* $0 < n < N_A$*.*

The theorem will be proved in Sec.4.2, and in Section 4.3 its compliance with cracking results will be shown by optimal attack and state-of-the-art cracking algorithm (i.e., PCFG [46]), respectively.

Corollary 1. *Suppose* $N_A \leq N_B, s_A \geq s_B$*. Then*

$$\lambda_A(n) \geq \lambda_B(n)$$

where $0 \leq n \leq N_B$ *(if* $n > N_A$*, define* $\lambda_A(n) = 1$*). If either inequalities of the above two conditions are strict, then* $\lambda_A(n) > \lambda_B(n)$*,* $0 < n < N_B$*.*

This corollary holds due to the evident fact that it is exactly the converse-negative proposition of Theorem 1.

The above theorem and corollary indicate that, given two password datasets A and B, we can first use liner regression to obtain their fitting lines (i.e., N_A, s_A, N_B and s_B), and then compare N_A with N_B, s_A with s_A, respectively: (1) If $N_A \geq N_B$ and $s_A \leq s_B$, dataset A is stronger than dataset B; (2) If $N_A \leq N_B$ and $s_A \geq s_B$, A is weaker than B; (3) For the remaining two cases where $N_A \geq N_B, s_A \geq s_B$ or $N_A \leq N_B, s_A \leq s_B$, the relationship between $\lambda_A(n)$ and $\lambda_B(n)$ depends on the discrete variable n, and thus it is generally unable to reach a conclusion and may have to resort to traditional methods (e.g., PCFG-based [46] or markov-based [33]) that are less accurate. Note that, in this work, datasets A and B may be from the same website but collected at different time points.

It should be noted that the metric proposed in this section can only be effective on password datasets that are in clear-text or un-salted hash. This restriction does not affect much its general applicability mainly for two reasons. On the one hand, our metric only involves offline operations to be performed after a relatively long period of time (e.g., a year), and thus the website can maintain one copy of salted passwords, which are online, to authenticate users and another copy of un-salted hash passwords, which are physically offline and well protected, to facilitate our measurement. On the other hand, we believe that websites with un-salted passwords are by no means a minority despite the difficulty to confirm this conjecture. The most convincing and obvious evidence lies in the fact that most of the previously leaked datasets from many prominent IT firms or leading organizations (such as Facebook, Adobe, Dropbox, IEEE, to name just a few [37]) are still in un-salted form. Furthermore, the authorities in many countries (e.g.,The National Security Agency of U.S. [31]) have been asking Internet providers and websites to provide user password datasets (in plain-text) to them. In this case, these websites shall also maintain a copy of un-salted passwords to ensure compliance with the regulations. In a nutshell, our metric is realistically practical.

4.2 Proof of the theorem

Obviously the theorem holds when $N_A = N_B, s_A = s_B$.

First we prove the theorem under the condition $s_A = s_B = s$, $N_A > N_B$. Recall that $f_r = \frac{C}{r^s}$, we denote the probability of a password with rank r be $p_r(= \frac{f_r}{sum} = \frac{C}{r^s \cdot sum})$. Then

$$\sum_{r=1}^{N_A} \frac{C_A}{r^s} = 1, \sum_{r=1}^{N_B} \frac{C_B}{r^s} = 1$$

$$C_A = \frac{1}{\sum_{r=1}^{N_A} \frac{1}{r^s}} < \frac{1}{\sum_{r=1}^{N_B} \frac{1}{r^s}} = C_B$$

So when $1 \le n \le N_B$, we have

$$\lambda_A(n) - \lambda_B(n) = (C_A - C_B)(\sum_{r=1}^{n} \frac{1}{r^s}) < 0$$

When $N_B + 1 \le n \le N_A - 1$,

$$\lambda_A(n) - \lambda_B(n) < 1 - 1 = 0$$

Next we prove the theorem under the conditions $N_A = N_B = N, s_A < s_B$

$$0 < C_A = \frac{1}{\sum_{r=1}^{N} \frac{1}{r^{s_A}}} < \frac{1}{\sum_{r=1}^{N} \frac{1}{r^{s_B}}} = C_B$$

When $1 \le n \le N - 1$,

$$\begin{aligned}
&\lambda_A(n) - \lambda_B(n) \\
=&\sum_{r=1}^{N} \frac{C_A}{r^{s_A}} - \sum_{r=1}^{N} \frac{C_B}{r^{s_B}} \\
=&C_A C_B (\sum_{r_1=1}^{N} \frac{1}{r_1^{s_B}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_A}} - \sum_{r_1=1}^{N} \frac{1}{r_1^{s_A}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_B}}) \\
=&C_A C_B (\sum_{r_1=1}^{n} \frac{1}{r_1^{s_B}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_A}} + \sum_{r_1=n+1}^{N} \frac{1}{r_1^{s_B}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_A}} - \sum_{r_1=1}^{n} \frac{1}{r_1^{s_A}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_B}} - \sum_{r_1=n+1}^{N} \frac{1}{r_1^{s_A}} \sum_{r_2=1}^{n} \frac{1}{r_2^{s_B}}) \\
=&C_A C_B (\sum_{1 \le r_2 \le n < r_1 \le N} (\frac{1}{r_1^{s_B} r_2^{s_A}} - \frac{1}{r_1^{s_A} r_2^{s_B}})) \\
=&C_A C_B (\sum_{1 \le r_2 \le n < r_1 \le N} \frac{1}{r_1^{s_A} r_2^{s_B}} ((\frac{r_1}{r_2})^{s_A - s_B} - 1))
\end{aligned}$$

For $r_1 > r_2, s_A < s_B$, so $(\frac{r_A}{r_2})^{s_A - s_B} < 1$. Further, we have

$$\lambda_A(n) - \lambda_B(n) < 0$$

Now the only left situation is $N_A > N_B, s_A < s_B$. We choose a password dataset C satisfying the conditions $N_C = N_A, s_C = s_B$, then

$$\begin{aligned}
\lambda_A(n) < \lambda_C(n) \quad 1 \le n \le N_A - 1 \\
\lambda_C(n) < \lambda_B(n) \quad 1 \le n \le N_A - 1
\end{aligned}$$

Thus $\lambda_A(n) < \lambda_B(n)$. This completes the proof.

Interestingly, we observe that, based on the conditions of Theorem 1, the three (preliminary) metrics (i.e., λ_β,μ_α, G_α) for assessing the strength of a given password dataset proposed by Bonneau [5] can be definitely compared with each other. Note that, λ_β stands for the success rate by β guesses under the optimal attack, μ_α stands for the least guesses needed to achieve a success rate of α, and G_α is used to measure the resistance to an online attack (or equally an offline guessing attack against salted passwords) and stands for the average guesses an attacker has to make in order to achieve a success rate of α by attacking every account at most μ_α times using an optimal strategy. Interested readers are referred to [5] for more details.

It is not difficult to see that λ_β is essentially the $\lambda^*(n)$ as in Eq.3, where β is analogous to n. According to Eq.4 and our Theorem 1, we get $\mu_\alpha(A) \ge \mu_\alpha(B)$. In addition,

$$
\begin{aligned}
G_\alpha =& (1-\lambda_{\mu_\alpha})\cdot\mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i = \sum_{i=1}^{\mu_\alpha}\sum_{j=1}^{i} p_i + (1-\lambda_{\mu_\alpha})\cdot\mu_\alpha \\
=& \sum_{j=1}^{\mu_\alpha}\sum_{i=j}^{\mu_\alpha} p_i + \sum_{j=1}^{\mu_\alpha}(1-\lambda_{\mu_\alpha}) = \sum_{j=1}^{\mu_\alpha}(1-\lambda_{\mu_\alpha} + \sum_{i=j}^{\mu_\alpha} p_i) \\
=& \sum_{j=1}^{\mu_\alpha}(1-\lambda_j)
\end{aligned}
$$

Since $\mu_\alpha(A) \geq \mu_\alpha(B)$ and $\lambda_j(A) \leq \lambda_j(B)$, we get

$$G_\alpha(A) \geq G_\alpha(B)$$

If either the two conditions in Theorem 1 is strict, then it holds that $G_\alpha(A) > G_\alpha(B)$, where $0 < \alpha \leq 1$.

4.3 Experimental results

In this subsection, we further use the simulated optimal attack and state-of-the-art password attacking algorithm (i.e., PCFG [23,46]) on real-life password datasets to demonstrate that our metric in Sec.4.1 is practically effective.

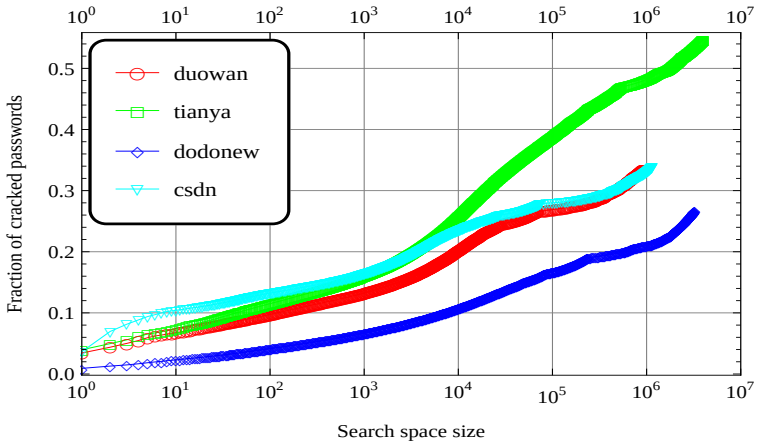
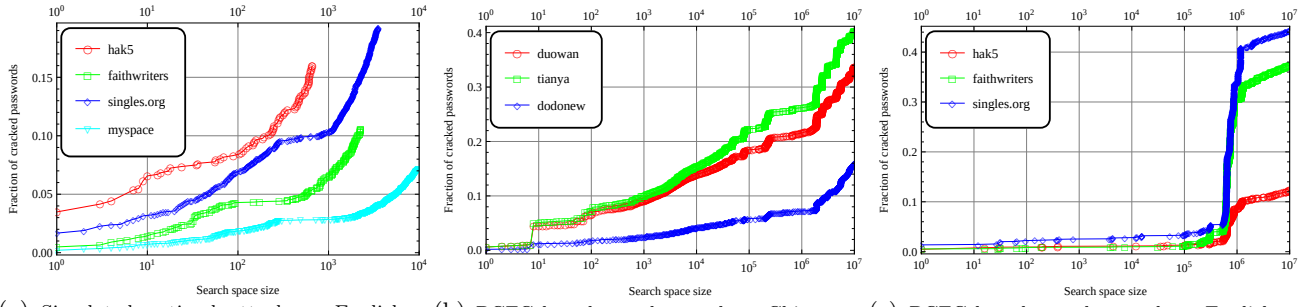


Fig. 5. Simulated optimal attack on four Chinese datasets (i.e., Tianya, Dodonew, Duowan and CSDN)



(a) Simulated optimal attack on English sets (1/4 training set against 1/4 test set)

(b) PCFG-based attacks on three Chinese datasets (using CSDN as the training set)

(c) PCFG-based attacks on three English datasets (using Myspace as the training set)

Fig. 6. Simulated optimal attacks and PCFG-based attacks on different groups of datasets

As the optimal attack is of theoretical importance to serve as the ultimate goal of any real attacks, it can by no means be seen as a realistic attack, for it assumes that the attacker is with *all* the plain-text passwords of the

target website. To see whether our metric accords with realistic attacks, we relax this assumption a bit and suppose that the attacker has obtained *a quarter of* the plain-text passwords of the target website (and use them to guess another quarter of the passwords, which is one-third of the remaining passwords). Note that this new assumption is much more realistic, because most of the compromised websites mentioned in this work have leaked a large part of their passwords in plain-text! And thus this new attacking scenario is rather practical and we call it "simulated optimal attack". For better presentation, we divide the eight datasets into two groups: one with dataset sizes all larger than one million and the other one smaller than one million. Simulated optimal attacking results on group one is illustrated in Fig.5, and results on group two is illustrated in Fig.6(a). It is not difficult to confirm that, for any two datasets in the same group, the attacking results comply with our metric results listed in Table 2. For instance, from Fig.5 we know that, for any search space size (i.e., ever n), dataset Tianya is weaker than dataset Duowan, this implies $N_{\text{tianya}} > N_{\text{duowan}}, s_{\text{tianya}} < s_{\text{duowan}}$. This implication accords with the statistics in Table 2.

Furthermore, we perform more realistic guessing attacks (i.e. PCFG-based attacks) to assess the effectiveness of our metric. As in simulated optimal attacks, we divide the eight datasets into two groups according to their sizes and user locations. For the Chinese group of datasets, we use CSDN as the PCFG training set and a corpus of wordlists (including the "english_lower.lst" from [15], "SogouLabDic.dic"[3] and the 20 million hotel reservations dataset[4]) as the input dictionary. The results are depicted in Fig.6(b). As for the English group of datasets, we use Myspace as the PCFG training set and a corpus of two wordlists (including the "dict-0294" [36] and "english_lower.lst" from [15]) as the input dictionary. The results are depicted in Fig.6(c).

The test shows that the PCFG-based attacking results on most of the datasets are consistent with our metric. As expected, there are leaps in the PCFG-based curves, while the simulated-attack-based curves are quite smooth. The reason is that the guess dictionary (in decreasing order) generated by PCFG are not as good as *suboptimal* (i.e., simulated optimal) guess dictionary – some guesses which should have been tested earlier are delayed, which further indicates PCFG-based attacks are far from *optimal*.

The only exception that violates our metric is on dataset Hak5. According to Table 2, N_{Hak5} is smaller than any other datasets and s_{Hak5} is larger than any other datasets in the same group, which means Hak5 is the weakest one. However, Fig.6(a) shows that, under the PCFG-based guessing attack, Hak5 is the strongest among the three English test sets. This inconsistence may be due to its non-representative nature of a real password dataset, or due to the inappropriateness of our selected training set and input dictionary.

Of particular interest may be our observation that, PCFG-based attacks seem to be much less effective than simulated optimal attacks. For example, at 1 million guesses, PCFG-based attacks on Chinese datasets achieve success rates 25%∼100% less than those of simulated optimal attacks. This gap is more pronounced for English datasets. It shouldn't come as a surprise, the gap in success rates and the aforementioned leaps in the PCFG-based curves are all due to the inherent weaknesses of PCFG-based attacks – their performance relies largely on the choices of training set and input dictionaries, while such choices are subject to too many uncertainties. This explains why we, in order to reach better success rates, divide our datasets into two groups according to populations and use different training sets and varied input dictionaries in our PCFG-based experiments. This also highlights the intrinsic limitations of using empirical attacking results (e.g., [23, 45]) as a strength measurement of password dataset. In a nutshell, there is still room for developing more practical attacking algorithms that have fewer uncertainties yet are more effective.

5 Conclusion

In this work, we have adopted techniques from computational statistics to demonstrate that the distribution of real-life passwords exactly obeys Zipf's law. Based on this observation, we put forward a novel metric to measure the strength of password datasets. Our metric achieves more accuracy and simplicity than existing metrics. The deterministic measurement of the password dataset strength provided by our metric facilitates system administrators to conduct fair and precise comparisons among different datasets. We formally proved our metric in a mathematically rigorous manner and also revealed some implications from Bonneau's α-guesswork [5]. We further evaluated the effectiveness of our metric by performing extensive experiments on a corpus of 56 million passwords and demonstrated its practicality.

More work remains to be done on this interesting yet challenging topic, as there are still many important problems that need to be investigated. For example, which kind of password creation policies tend to lead to more secure passwords? Is it user-acceptable to employ a password creation policy like Schechter et al. [39] that only

[3] http://www.sogou.com/labs/dl/w.html
[4] http://www.4hoteliers.com/news/story/12047

allows passwords with popularity lower than a threshold (e.g., five times)? There are also many new issues brought about by the findings in this work. For instance, what are the implications of the Zipf's law in passwords for the cryptographic protocol community? As most of the existing password authenticated key exchange (PAKE) protocols (e.g., [3, 22]) have been proved secure in some security model (i.e., the random oracle model or the standard model) under the hypothesis that passwords are uniformly distributed, do such security results still hold under the realistic distribution of real-life passwords? We believe this paper will extend the results from password analyses to new insights into password-based cryptographic protocols.

References

1. Alsaleh, M., Mannan, M., Van Oorschot, P.: Revisiting defenses against large-scale online password guessing attacks. IEEE Trans. Dependable and Secure Computing 9(1), 128–141 (2012)
2. Axtell, R.L.: Zipf distribution of US firm sizes. Science 293(5536), 1818–1820 (2001)
3. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: Preneel, B. (ed.) Proc. EUROCRYPT 2000, LNCS, vol. 1807, pp. 139–155. Springer Berlin/Heidelberg (2000)
4. Bishop, M., V Klein, D.: Improving system security via proactive password checking. Computers & Security 14(3), 233–249 (1995)
5. Bonneau, J.: The science of guessing: Analyzing an anonymized corpus of 70 million passwords. In: Proc. 33th IEEE Symp. on Security and Privacy. pp. 538–552. IEEE (2012)
6. Bonneau, J., Herley, C., Oorschot, P., Stajano, F.: The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In: Proc. 33th IEEE Symp. on Security and Privacy. pp. 553–567. IEEE (2012)
7. Bowes, R.: Password dictionaries (Oct 2011), `https://wiki.skullsecurity.org/Passwords`
8. Burr, W., Dodson, D., Perlner, R., Polk, W., Gupta, S., Nabbus, E.: NIST SP800-63-2 – electronic authentication guideline. Tech. rep., NIST, Reston, VA (Aug 2013)
9. de Carnavalet, X.d.C., Mannan, M.: From very weak to very strong: Analyzing password-strength meters. In: Proc. NDSS 2014 (23-26 Feb 2014)
10. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from markov models. In: Proc. NDSS 2012. pp. 1–15 (2012)
11. Chiasson, S., Stobert, E., Forget, A., Biddle, R., Van Oorschot, P.C.: Persuasive cued click-points: Design, implementation, and evaluation of a knowledge-based authentication mechanism. IEEE Trans. on Dependable and Secure Computing 9(2), 222–235 (2012)
12. Constantin, L.: Security Gurus 0wned by Black Hats (July 2009), http://news.softpedia.com/news/Security-Gurus-0wned-by-Black-Hats-117934.shtml
13. Davies, C., Ganesan, R.: Bapasswd: A new proactive password checker. In: 16th National Computer Security Conference. pp. 1–15 (1993)
14. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: Proc. INFOCOM 2010. pp. 1–9. IEEE (2010)
15. Designer, S.: John the Ripper password cracker (Feb 1996), `http://www.openwall.com/john/`
16. Dürmuth, M.: Useful password hashing: how to waste computing cycles with style. In: Proc. NSPW 2013. pp. 31–40. ACM (2013)
17. Furusawa, C., Kaneko, K.: Zipfs law in gene expression. Physical review letters 90(8), 088102 (2003)
18. Herley, C., Van Oorschot, P.: A research agenda acknowledging the persistence of passwords. IEEE Security & Privacy 10(1), 28–36 (2012)
19. Houshmand, S., Aggarwal, S.: Building better passwords using probabilistic techniques. In: Proc. ACSAC 2012. pp. 109–118. ACM (2012)
20. Inglesant, P.G., Sasse, M.A.: The true cost of unusable password policies: password use in the wild. In: Proc. of 28th ACM Conference on Human Factors in Computing Systems (CHI 2010). ACM (2010)
21. Katalov, V.: Yahoo!, Dropbox and Battle.net Hacked: Stopping the Chain Reaction (Feb 2013), http://blog.crackpassword.com/2013/02/yahoo-dropbox-and-battle-net-hacked-stopping-the-chain-reaction/
22. Katz, J., Ostrovsky, R., Yung, M.: Efficient and secure authenticated key exchange using weak passwords. J. ACM 57(1), 1–41 (2009)
23. Kelley, P.G., Komanduri, S., Mazurek, M.L., Shay, R., Vidas, T., Bauer, L., Christin, N., Cranor, L.F., Lopez, J.: Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In: Proc. IEEE S&P 2012. pp. 523–537. IEEE (2012)
24. Klein, D.V.: Foiling the cracker: A survey of, and improvements to, password security. In: Proc. of the 2nd USENIX Security Workshop. pp. 5–14 (1990)
25. Kuo, C., Romanosky, S., Cranor, L.F.: Human selection of mnemonic phrase-based passwords. In: Proc. SOUPS 2006. pp. 67–78. ACM (2006)
26. Long, J.: No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing. Syngress (2011)

27. Ma, J., Yang, W., Luo, M., Li, N.: A study of probabilistic password models. In: Proc. IEEE S&P 2014. pp. 1–16. IEEE (2014)
28. Maillart, T., Sornette, D., Spaeth, S., Von Krogh, G.: Empirical tests of zipf's law mechanism in open source linux distribution. Physical Review Letters 101(21), 218701 (2008)
29. Malone, D., Maher, K.: Investigating the distribution of password choices. In: Proc. WWW 2012. pp. 301–310. ACM (2012)
30. Martin, R.: Amid Widespread Data Breaches in China (Jan 2012), https://sg.finance.yahoo.com/news/Amid-Widespread-Data-Breaches-pennolson-706259476.html
31. McCullagh, D.: Feds tell Web firms to turn over user account passwords (July 2013), http://www.cnet.com/news/feds-tell-web-firms-to-turn-over-user-account-passwords
32. Morris, R., Thompson, K.: Password security: A case history. Communications of the ACM 22(11), 594–597 (1979)
33. Narayanan, A., Shmatikov, V.: Fast dictionary attacks on passwords using time-space tradeoff. In: Proc. CCS 2005. pp. 364–372. ACM (2005)
34. Nievergelt, Y.: Total least squares: State-of-the-art regression in numerical analysis. SIAM review 36(2), 258–264 (1994)
35. Oechslin, P.: Making a faster cryptanalytic time-memory trade-off. In: Proc. CRYPTO 2003. pp. 617–630. Springer (2003)
36. Outpost9.com's Lab: Word lists (Feb 2014), `http://www.outpost9.com/files/WordLists.html`
37. Prerad, M.: 20 Biggest Data Breaches of 2013, https://www.linkedin.com/today/post/article/20140224 081155-67886711-20-biggest-data-breaches-of-2013
38. Rao, A., Jha, B., Kini, G.: Effect of grammar on security of long passwords. In: Proc. of CODASPY 2013. pp. 317–324. ACM (2013)
39. Schechter, S., Herley, C., Mitzenmacher, M.: Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks. In: Proc. HotSec 2010. pp. 1–8 (2010)
40. Shirvanian, M., Jarecki, S., Saxena, N., Nathan, N.: Two-factor authentication resilient to server compromise using mix-bandwidth devices. In: Proc. NDSS 2014. pp. 1–16 (Feb 23–26 2014)
41. Spafford, E.: Observations on reusable password choices. In: Proc. USENIX Security Workshop (1992)
42. Spafford, E.H.: Opus: Preventing weak password choices. Computers & Security 11(3), 273–278 (1992)
43. Ur, B., Kelley, P.G., Komanduri, S., Lee, J., Maass, M., Mazurek, M., Passaro, T., Shay, R., Vidas, T., Bauer, L., et al.: How does your password measure up? the effect of strength meters on password creation. In: Proc. USENIX Security 2012 (Feb 2012)
44. Verheul, E.R.: Selecting secure passwords. In: Abe, M. (ed.) Proc. CT-RSA 2007, LNCS, vol. 4377, pp. 49–66. Springer Berlin / Heidelberg (2007)
45. Weir, M., Aggarwal, S., Collins, M., Stern, H.: Testing metrics for password creation policies by attacking large sets of revealed passwords. In: Proc. CCS 2010. pp. 162–175. ACM (2010)
46. Weir, M., Aggarwal, S., de Medeiros, B., Glodek, B.: Password cracking using probabilistic context-free grammars. In: Proc. 30th IEEE Symp. on Security and Privacy. pp. 391–405. IEEE (2009)
47. Wu, T.: A real-world analysis of kerberos password security. In: Proc. NDSS 1999. pp. 13–22 (1999)
48. Yan, J.J., Blackwell, A.F., Anderson, R.J., Grant, A.: Password memorability and security: Empirical results. IEEE Security & privacy 2(5), 25–31 (2004)
49. Zhang, Y., Monrose, F., Reiter, M.K.: The security of modern password expiration: an algorithmic framework and empirical analysis. In: Proc. CCS 2010. pp. 176–186. ACM (2010)
50. Zipf, G.K.: Human behavior and the principle of least effort. Addison-Wesley Press (1949)