Zipf's Law in Passwords

Ding Wang[1], Gaopeng Jian[2], Ping Wang[1]

[1] School of EECS, Peking University, Beijing 100871, China
[2] School of Mathematical Sciences, Peking University, Beijing 100871, China
wangdingg@mail.nankai.edu.cn

Abstract. Despite more than thirty years of intensive 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. By conducting linear regressions on a corpus of 56 million passwords, we for the first time show that Zipf's law *perfectly* exists in user-generated passwords, figure out the corresponding exact distribution functions, and investigate some *fundamental* implications of our observations for password policies and password-based authentication.

As one specific application of this law of nature, we propose the number of unique passwords used in regression and the absolute value of slope of the regression line together as a metric for assessing the strength of password datasets, and prove it in a mathematically rigorous manner. In addition, 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. Our new metric further develops Bonneau's α-guesswork and to the best of knowledge, it 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: Internet security, User authentication, Passwords, Zipf's law, Probabilistic context-free grammar

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., multi-factor authentication [30], graphical passwords [17]) 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 [11] and moreover, the transition costs of replacing them can not be effectively quantified [28], 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 [74]. 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 [10] 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 [74], rule-based [9, 59], entropy-based [13, 66] and cracking-based [15, 29]) 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 [14], 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 [14]) 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 [65]. Inglesant and Sasse [31] 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, they will try every means to circumvent 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

As different systems favor varied password policies, now some natural questions arise: how can the password policy designers evaluate their policies? How can the administrators prefer the right policy for their systems? In addition, usually the services provided by a system and its users may vary as time goes on, which may lead to huge changes in the password dataset after some period of time (e.g., one year) even though the password policy[3] stays the same. In this situation, the system administrators 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 core crux of designing and preferring an appropriate password policy or properly adjusting it lies in how to accurately assess the strength of password datasets that are created under it. Note that, in this work we presume that a state-of-the-art password creation policy (e.g., [15, 29]) 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 two most commonly used approaches to assess the strength of a given password dataset are estimating its information entropy (e.g., [13]) and empirically analyzing its "guessability" (e.g., [37, 71]). The former, however, is not based on empirical data and has been shown inaccurate [71], and the latter, which largely depends on the choices of the cracking algorithms and input dictionaries [20, 43], has too many uncertainties to accurately characterize the strength of a given dataset. Very recently, Bonneau [10] 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.

The settlement of the above question will naturally entail the settlement of a more fundamental question as the pre-condition: How to precisely characterize a given password dataset? Or equally, what's the distribution of passwords? Despite more than thirty years of intensive research efforts, textual passwords are still enveloped in mysterious veils and the same old question is asked year in year out, which may well explains why most password authenticated key exchange (PAKE) protocols with provable security (some notable ones include [7, 16, 55] in the random oracle model and [27,35,36] in the standard model) generally rely on a simple but inconceivable assumption: Passwords follow a uniform distribution. To the best of our knowledge, Malone and Maher's work [45] may be the most relevant to what we will discuss in the current paper. They made an attempt to investigate 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 their 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 rigorous answers to the above-mentioned two interesting (and important) questions: (1) *What's the exact distribution of (user-generated) passwords?* and (2) *How to accurately measure the security strength of a given password dataset?*

As our primary contribution, we 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, such as such as the Internet topology [24] and US firm sizes [4]. We prune these least frequent passwords and rank the frequency of each remaining unique password in decreasing order

[3] Note that, in this work the terms "policy", "password policy" and "password creation policy" will be used interchangeably.

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, user localization and language (culture background), 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 [45]. We further investigate the general applicability of our observation and highlight its fundamental implications for password policies and password-based cryptographic protocols (e.g., authentication, signature and secret sharing). This provides a compelling answer to the first question above.

Our second contribution is a novel metric that facilitates the password policy designers and system administrators to have a concise grasp of the strength of their password datasets (either in plain-text or hashed form) in a mathematically rigorous manner, and enable them to precisely evaluate the security property of password policies under examination. For instance, the system administrators may compare the password-dataset strength 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), and based on these comparison results, they 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 second question.

Our final 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 then take a step forward to employ the state-of-the-art cracking algorithm (i.e., probabilistic context-free grammars (PCFG) [37, 72]) 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 [10] 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 5. Section 6 conclues the paper.

2 Related Work

In this section, we briefly review some related works on password creation policies and password cracking techniques to provide some backgrounds for later discussions.

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" [38]. 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 [62] and BApasswd [19]. 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 [13]. However, by modeling the success rates of current password cracking techniques against real-life user passwords under different rules, Weir et al. [71] 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 [71], Houshmand et al. [29] proposed a novel policy that first analyzes whether a user selected password is weak or strong according to empirical PCFG training 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 within a continuous range.

Perhaps the most relevant policy related to our strength metric for assessing password datasets (see Sec.5) is suggested by Schechter et al. [59]. 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, this policy rejects passwords that occur at a probability exceeding a threshold r (e.g., $r = \frac{1}{10^6}$ as exampled in their work [59]), yet whether it would greatly reduce usability has not been evaluated thoroughly (e.g., no actual use case studies are reported). As one can see, an immediate consequence of this policy is that, it would frequently frustrate users to use their intended passwords that are typically 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 select and maintain a new password. In all, such a policy would be very promising if these issues can be addressed.

2.2 Password cracking

Password-based systems are prone to various attacks, including on-line guessing, offline guessing, keylogging and social engineering [42]. 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 [3], 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 (e.g., using PBKDF2 [64]) 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 [20]. For example, system countermeasures against offline guessing attacks, such as salting to defeat pre-computation techniques (e.g., Rainbow tables [52]) or key strengthening techniques [22] 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 [53]. 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 [53], and this sort of techniques have emerged as early as 1979 in Morris-Thompson's analysis of 3,000 passwords [48]. This initial work has been followed by independent works by Klein [38] and Spafford [61]. Later on, some dedicated software tools like John the Ripper [21] appeared. Subsequent studies (e.g., [40, 73]) 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 [49] 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. [72] 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 [21]. In 2010, Zhang et al. [76] found Weir et al.'s algorithm is the most effective one among all techniques (including Markov model [49], John the Ripper [21] and Rainbow [52]) they used, which has also been confirmed by [10, 15]. 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

In this section, we first give some background on the statistical technique—linear regression—used in the following, and then briefly describe the collected datasets. Further, we provide a fundamental understanding of passwords and show that Zipf's law perfectly exists in real-life passwords. Finally, we discuss some foundational implications of our findings.

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, given the value of x, the conditional mean of y 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 [51]. This method computes the best-fitting line for the observed data by minimizing the sum of the squares of the vertical deviations from each 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 the password datasets

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

All eight datasets, as summarized in Table 1, were compromised by hackers or leaked by anonymous insiders, and were subsequently disclosed publicly on the Internet. They have also been used by a number of scientific works that study passwords (e.g., [32, 41, 43, 71]). We realize that while publicly available, these datasets contain private data such as emails, user names and passwords. Therefore, we treat all user names as confidential and only report the aggregation information about passwords such that using them in our research does not increase the harm to the victims. Furthermore, attackers are likely to exploit these accounts as training sets or cracking dictionaries, while our study of them are of practical relevance to security administrators and common users to secure their accounts.

Table 1. Basic information about the eight datasets

Dataset	Application	Location	Language	When leaked	How leaked	Total Passwords	Unique Passwords
Tianya	Social forum	China	Chinese	Dec. 4, 2011	Hacker breached	30,233,633	12,614,676
Dodonew	Gaming&E-commerce	China	Chinese	Dec. 3, 2011	Hacker breached	16,231,271	11,236,220
CSDN	Programmer forum	China	Chinese	Dec. 2, 2011	Hacker breached	6,428,287	4,037,610
Duowan	Gaming	China	Chinese	Dec. 1, 2011	Insider disclosed	4,982,740	3,119,070
Myspace	Social forum	USA	English	Oct. 1, 2006	Social engineering (phishing)	41,545	37,144
Single.org	Dating	USA	English	Oct. 1, 2010	Query string injection	16,250	12,234
Faithwriters	Writer forum	USA	English	Mar. 1, 2009	SQL injection	9,709	8,347
Hack5	Hacker forum	USA	English	July 1, 2009	Hacker breached	2,987	2,351

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 [12] 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) [18]. 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.3.4) 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 Tianya, Dodonew, CSDN and Duowan, 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 [46] and we collected them at that time. CSDN is the largest community website of Chinese programmers; Tianya is an influential Chinese BBS; Duowan is a popular game forum; Dodonew is also a popular game forum and it enables monetary transactions. 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.

3.3 Statistics about the password datasets

In this subsection, we report some statistical information about our datasets. Firstly, the character composition information is summarized in Table 2. 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, Myspace users prefer to generate their passwords by adding the digit "1" to a sequence of lower-case letters.

Table 2. Character composition information about each password dataset

Dataset	Total 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	9.96%	0.18%	10.29%	**63.77%**	98.05%	14.63%	0.12%	15.64%	4.37%	4.11%
Dodonew	16,231,271	8.79%	0.27%	9.37%	**20.49%**	82.88%	40.81%	1.39%	42.94%	7.31%	6.95%
CSDN	6,428,287	11.64%	0.47%	12.35%	**45.01%**	96.31%	26.14%	0.24%	28.45%	6.46%	5.88%
Duowan	4,982,740	10.30%	0.09%	10.52%	**52.84%**	97.59%	23.97%	0.37%	24.84%	6.04%	5.83%
Myspace	41,545	7.18%	0.31%	7.66%	0.71%	89.95%	**65.66%**	**18.24%**	**69.77%**	6.02%	5.66%
Singles.org	16,250	60.20%	1.92%	**65.82%**	9.58%	99.78%	17.77%	2.73%	19.68%	1.92%	1.77%
Faithwriters	9,709	54.40%	1.16%	**59.04%**	6.35%	99.57%	22.82%	4.13%	25.45%	2.73%	2.37%
Hak5	2,987	18.61%	0.27%	**20.39%**	5.56%	92.13%	16.57%	2.01%	31.80%	1.44%	1.21%

Table 3 demonstrates the length distributions of each dataset. We can see that the most popular password lengths are 6 to 10, which on average account for 85.01% of the whole dataset. Few users choose passwords that are longer than 12, with Dodonew being an exception. One plausible reason may be that, www.dodonew.com is a website that enables monetary transactions and its users perceive their accounts as being important, and thus longer passwords are selected. Of particular interest to our observations is that the CSDN dateset has much fewer length 6 and 7 passwords as compared to other datasets. This may be due to the fact that www.csdn.net (as well as many other websites) started with a loose password policy and later on a strict policy was enforced (e.g., requiring the passwords to be of a minimum-8 length). We also note that passwords from www.christian-singles.org are all no longer than 8, which may be due to a policy that prevents users from choosing passwords longer than 8 characters. Such a policy still exits in many financial companies [33], and a plausible reason may be that the shift to longer allowed password lengths is a non-trivial issue.

Table 3. Length distribution information of each dataset

Length	1-3	4	5	6	7	8	9	10	11	12	13-16	17-30	30+	All
Tianya	0.61%	0.65%	0.55%	33.77%	13.92%	18.10%	9.59%	10.28%	5.53%	2.88%	4.05%	0.07%	0.00%	100%
Dodonew	0.36%	0.70%	0.78%	9.71%	13.45%	18.49%	20.29%	14.69%	3.10%	1.34%	10.24%	6.79%	0.04%	100%
CSDN	**0.01%**	**0.10%**	**0.51%**	**1.29%**	**0.26%**	36.38%	24.15%	14.48%	9.78%	5.75%	6.96%	0.32%	0.00%	100%
Duowan	0.02%	0.13%	0.12%	20.62%	17.68%	22.49%	15.12%	11.55%	5.30%	2.72%	4.13%	0.12%	0.00%	100%
Myspace	0.25%	0.51%	0.79%	15.67%	23.40%	22.78%	17.20%	13.65%	2.83%	1.13%	1.15%	0.48%	0.17%	100%
Singles.org	0.68%	4.74%	7.68%	32.05%	23.20%	31.65%	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	**0.00%**	100%
Faithwriters	0.04%	0.14%	0.99%	31.97%	20.95%	22.71%	10.35%	5.98%	3.24%	1.87%	1.53%	0.20%	0.01%	100%
Hak5	0.10%	0.64%	0.97%	12.96%	8.50%	20.89%	8.94%	30.83%	3.58%	3.08%	6.90%	2.44%	0.17%	100%
Average	0.26%	0.95%	1.55%	**19.75%**	**15.17%**	**24.19%**	**13.20%**	**12.68%**	4.17%	2.35%	4.37%	1.30%	0.05%	100%

In the 1980s, it was revealed that the most popular password at that time was `12345`; thirty years later, as can be seen from Table 4, `123456` takes the lead. Our results accord with the statistics given in [67]. It is a long-standing problem that a significant fraction of users prefer the same passwords as if by prior agreement, and this is in part due to the inherent limitations of human cognition. Note that, this situation can not be fundamentally altered by simply banning such popular passwords. For example, if `password` is banned, then `password1` will be popular (see the most popular passwords of Myspace); if `password1` is banned, then `pa$$word1` will be popular. It is hoped that the adaptive password meters (e.g., [15, 29]) will ultimately eliminate this issue. After having examined Table 2, it is expected to see that most of the top 10 popular passwords of Chinese users are sole digits, while most of the top 10 popular passwords of English users are sole letters. What's interesting is that "love" is also the eternal theme of passwords: five datasets have a most popular password related to "love". For instance, the password `5201314`, which sounds as "I love you forever and ever" in Chinese, ranks the 5*th* and 7*th* most popular password in Dodonew and Tianya, respectively. It is alarming to see that for several datasets, a mere of top 3 most popular passwords account for more than 5% of all the passwords, which indicates that, to break into these corresponding websites, an online (trawling) guessing attacker will succeed every one in twenty attempts. Also, as a side note, even though popular passwords in Hak5 look rather complex (diversified) and actually about 66.18% of its passwords are composed of a mixture of lower/upper-case letters and numbers, this dataset is still very concentrated and as we will show later in Section 5, it is among the weakest ones. This means that seemingly complex passwords may not actually be difficult to crack, which further suggests the necessity and importance of a fundamental understanding of passwords.

Table 4. Top 10 most popular passwords of each dataset

Rank	Tianya	Dodonew	CSDN	Duowan	Myspace	Singles.org	Faithwriters	Hak5
1	**123456**	**123456**	123456789	**123456**	password1	**123456**	**123456**	QsEfTh22
2	111111	a123456	12345678	111111	abc123	jesus	writer	——
3	000000	123456789	11111111	123456789	fuckyou	password	jesus1	timosha
Percent of top3	5.58%	1.49%	8.15%	5.01%	0.40%	2.10%	1.03%	4.62%
4	123456789	111111	dearbook	123123	monkey1	12345678	christ	ike02banaA
5	123123	**5201314**	00000000	000000	**iloveyou1**	christ	blessed	**123456**
6	123321	123123	123123123	**5201314**	myspace1	**love**	john316	zxczxc
7	**5201314**	a321654	1234567890	123321	fuckyou1	**princess**	jesuschrist	123456789
8	12345678	12345	88888888	a123456	number1	jesus1	password	westside
9	666666	000000	111111111	suibian	football1	sunshine	heaven	ZVjmHgC355
10	111222tianya	123456a	147258369	12345678	nicole1	1234567	faithwriters	Kj7Gt65F
Percent of top10	7.42%	3.28%	10.44%	6.78%	0.78%	3.40%	2.17%	7.20%

3.4 Zipf's law in passwords

Initially, probabilistic context-free grammar (PCFG) is a machine learning technique used in natural language processing (NLP), yet Weir et al. [72] 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. [57].

Inspired by these earlier works, in this study for the first time we attempt to investigate whether the Zipf's law,[4] 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 [77]. 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 [44], gene expression [25], as well as US firm sizes [4].

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

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

[4] Zipf's law distributions are also called Pareto or power-law distributions, and they are different ways of looking at the same thing—all can be derived from each other [2].

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 $\log(r)$:

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

Table 5. Linear regression results of password datasets ("PWs" stands for passwords)

Dataset	Totoal PWs	Least frequency	Total PWs in regression	Fraction of PWs in regression (θ)	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

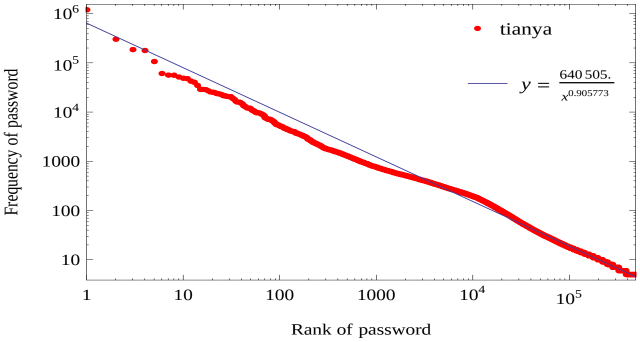


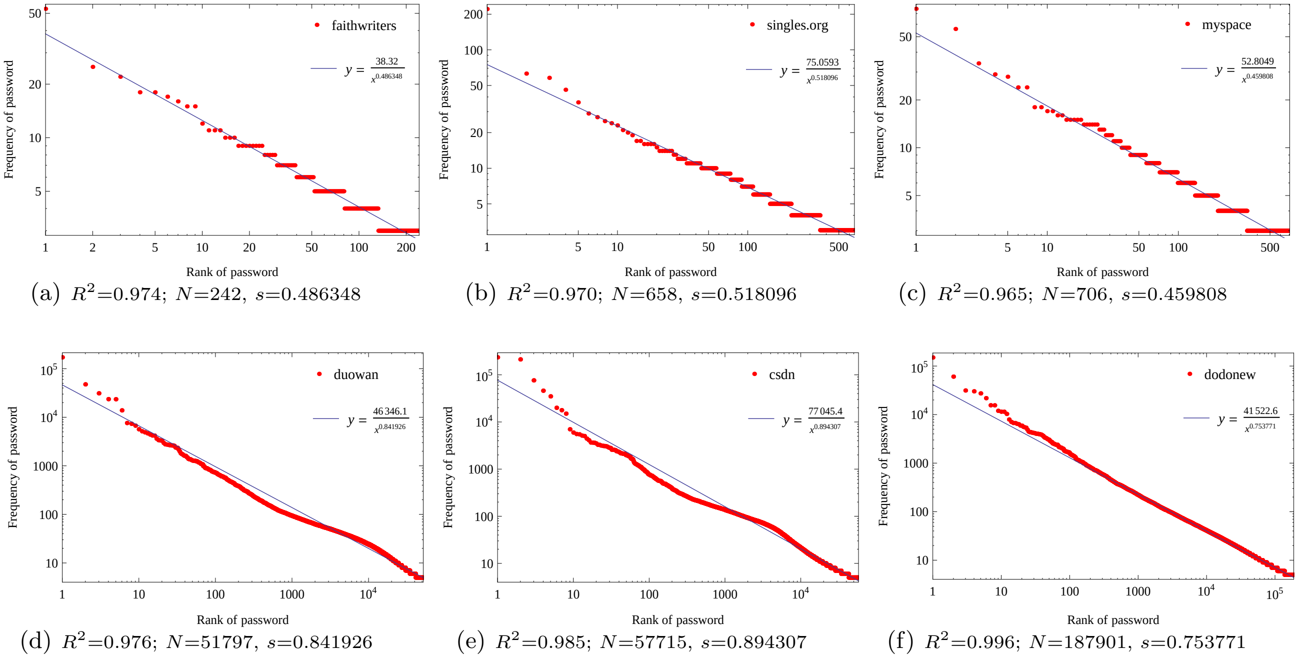
Fig. 1. Zipf's law in Tianya ($R^2 = 0.994$)

As can be seen from Fig.1, 30 million passwords from the website www.tianya.cn conform to Zipf's law to such an 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 lines well represent 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 5, 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 5, datasets leaked by phing attacks are likely to have a lower R^2 as compared to that of datasets leaked by website breaches, for phing attacks are unlikely to obtain the entire dataset of a website, while website breaches, 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, the 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



(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

dataset. This conjecture will be established by both rigorous proofs and extensive empirical experiments in the following section. In addition, to qualify as a proper description of a dataset, a distribution function $f(x)$ shall hold within a range $x_{min} \le f(x) \le x_{max}$ of at least 2 $\sim$ 3 orders of magnitude (i.e., $x_{max}/x_{min} \ge 10^{2 \sim 3}$) [44], and one can see that this condition is satisfied in all our regressions. Particularly, it is the popular passwords (i.e., *the front head* of the whole passwords) that perfectly follow a Zipf's law.

We note that Malone and Maher [45] have also attempted to investigate the distributions of real-life passwords, yet contrary to our findings, they 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." We figure out the primary cause of their failure–they fitted all the passwords of a dataset to the Zipf model. More specifically, unpopular passwords (e.g., with frequency less than five) are extremely common (see Table 5) and constitute the long tail (see Fig.1 of [45] for a concise grasp),[5] yet they fail to reflect their true popularity due to the veiled fact that all datasets used in regression are not sufficiently large enough, even though some datasets are in millions. For example, www.csdn.net adopts a policy that passwords shall be composed of letters and numbers and of length with a minimum-8 and maximum-16, this means that a user's password (denoted by a stochastic variable X) will have about $|X| = 62^{16} - 62^{8}$ possible (distinct) values under this policy, but we have only got 6.42 million passwords from CSDN, a very small sample as compared to $|X|$. Even though the variable X perfectly obeys Zipf's law, a small sample without proper data processing is highly unlikely to reflect this nature. As a result, it is definite that the regression process will be largely affected by those unpopular passwords and no marked rule can be observed even if the front head of passwords exhibits a good property.

To further justify the need for excluding unfrequent passwords in the data processing, we investigate three parameters, i.e., exact distribution (3 kinds), sample size (7 kinds) and the least frequency concerned (5 kinds, denoted by LF), that might influence a regression and thus perform a series of $105(= 3 \cdot 5 \cdot 7)$ experiments. More specifically, suppose that the stochastic variable X follows the Zipf's law and there are N =1000 possible values $\{x_1, x_2, \cdots, x_{1000}\}$ for X. Without loss of generality, the distribution law is defined to be $\{p(x_1) = \frac{C/1^s}{\sum_{i=1}^{N} \frac{C}{i^s}} = \frac{1/1^s}{\sum_{i=1}^{N} \frac{1}{i^s}}$, $p(x_2) = \frac{C/2^s}{\sum_{i=1}^{N} \frac{C}{i^s}} = \frac{1/2^s}{\sum_{i=1}^{N} \frac{1}{i^s}}, \cdots, p(x_N) = \frac{C/N^s}{\sum_{i=1}^{N} \frac{C}{i^s}} = \frac{1/N^s}{\sum_{i=1}^{N} \frac{1}{i^s}}\}$, where the sample space N and the slope s determine the exact Zipf distribution function. To be robust, each experiment is run 1000 times and with only one parameter changing. Due to space constraints, Table 6 only incorporates 35 experiments where s is fixed to 0.9, the sample size varies from 100 to 10000 and LF increases progressively from 1 to 5. Readers are referred to the full results

[5] It is also called the "noisy tail" [50] in the statistical domain.

about all 105 experiments in [68]. Note that some integral statistics (e.g., the fitted N) in Table 6 are with decimals because they are averaged over 1000 repeated experiments.

Table 6. Effects of sample size and least frequency on linear regression when simulating a Zipf distribution

Zipf N	Zipf s	Sample size	Least frequency(LF)	# of Unique passwords	Passwords used in regression	Passwords used in regression(%)	Fitted N	Fitted s	R^2
1000	0.9	100	**1**	71.197	100.000	100.00%	71.197	0.429486	0.754566
1000	0.9	100	2	71.262	41.099	41.10%	12.361	0.641264	0.884263
1000	0.9	100	3	70.963	27.201	**27.20%**	5.307	**0.719897**	0.894042
1000	0.9	100	4	71.068	20.585	20.59%	3.173	0.683547	0.916477
1000	0.9	100	5	70.765	17.010	17.01%	2.215	0.622484	0.953243
1000	0.9	200	1	123.933	200.000	100.00%	123.933	0.516278	0.822066
1000	0.9	200	2	124.103	102.971	51.49%	27.074	0.688394	0.923847
1000	0.9	200	3	123.795	73.429	36.71%	12.145	0.761613	0.935451
1000	0.9	200	4	124.121	59.139	**29.57%**	7.392	**0.785336**	**0.930795**
1000	0.9	200	5	123.954	50.151	25.08%	5.242	0.784747	0.921241
1000	0.9	500	1	245.459	500.000	100.00%	245.459	0.633549	0.895852
1000	0.9	500	2	246.040	326.859	65.37%	72.899	0.724630	0.951529
1000	0.9	500	3	245.482	250.498	50.10%	34.245	0.796940	0.969880
1000	0.9	500	4	245.697	211.680	42.34%	21.499	0.819386	0.970288
1000	0.9	500	5	245.586	187.536	**37.51%**	15.372	**0.834885**	**0.966581**
1000	0.9	1000	1	389.360	1000.000	100.00%	389.36	0.730031	0.937941
1000	0.9	1000	2	388.014	760.039	76.00%	148.053	0.756649	0.965318
1000	0.9	1000	3	388.733	611.795	61.18%	74.478	0.807381	0.979783
1000	0.9	1000	4	388.774	530.803	53.08%	47.184	0.833071	0.983395
1000	0.9	1000	5	388.839	476.921	**47.69%**	33.829	**0.847137**	**0.983550**
1000	0.9	2000	1	573.821	2000.000	100.00%	573.821	0.835995	0.964407
1000	0.9	2000	2	573.607	1712.451	85.62%	286.058	0.790817	0.977339
1000	0.9	2000	3	574.446	1455.076	72.75%	158.041	0.818059	0.985691
1000	0.9	2000	4	574.011	1287.865	64.39%	102.03	0.840089	0.989460
1000	0.9	2000	5	574.229	1173.160	**58.66%**	73.534	**0.854452**	**0.990812**
1000	0.9	5000	1	828.243	5000.000	100.00%	828.243	0.963949	0.963691
1000	0.9	5000	2	828.466	4760.094	**95.20%**	588.56	**0.861714**	**0.989008**
1000	0.9	5000	3	827.675	4379.226	87.58%	397.276	0.842637	0.991843
1000	0.9	5000	4	828.601	4014.673	80.29%	276.308	0.849865	0.993588
1000	0.9	5000	5	828.281	3724.258	74.49%	203.349	0.859765	0.994832
1000	0.9	10000	1	953.483	10000.000	100.00%	953.483	1.013698	0.943442
1000	0.9	10000	2	953.545	9884.596	98.85%	838.141	0.929787	0.985080
1000	0.9	10000	3	953.125	9582.080	**95.82%**	686.791	**0.884120**	**0.994655**
1000	0.9	10000	4	953.483	9146.947	91.47%	541.471	0.867965	0.996179
1000	0.9	10000	5	953.365	8683.549	86.84%	425.614	0.866388	0.996641

Our intensive results shows that, given a Zipf distribution (i.e., N and s is fixed), no matter the size of sampled data is smaller, equal or larger than N, larger LF will lead to a better regression (i.e., the fitted s is closer to the fixed s and R^2 closer to 1) in the beginning and but would worsen the situation as LF further increases. More specifically, when the size of sampled data is smaller than N, the fitted s first increases and then decreases as LF increases progressively; When the size of sampled data is larger than N, on the contrary, the fitted s first decreases and then increases as LF increases progressively. Since the sizes of real-life password datasets are generally smaller than the password sample space, it is reasonable and necessary to prune these least frequent passwords when performing regression. This well explicates why diametrically opposed conclusions are reached between [45] and this work.

3.5 General applicability of our observations

In the regressions in the previous section, we have only considered datasets that are generated under loose password composition policies. As can be seen from Table 1∼3, quite short and letter-only passwords exist in all eight datasets, which suggests that there is no evident length or composition requirement for generating passwords in any of these eight websites. We believe a more precise and reasonable explanation for this phenomenon is that most of these passwords are created under a mixture of unknown policies: Initially, there is no rule (policy); Later on, some (loose or strict) rule(s) is applied; Sometime later, the sites were hacked and passwords disclosed.

However, this is not the case in many cases, especially for security-critical applications which may implement strict policies at the very start. To further demonstrate the applicability of our findings, two special kinds of datasets created under more constrained (yet quite realistic) password composition policies are considered: (1) Datasets with password lengths no shorter than some minimum length (e.g., at least eight characters long); and (2) Datasets with each password including a mix of letters and numbers (e.g., at least one letter and one number).

Since we did not have exact examples of passwords exactly generated under some specific creation policy with a length or composition requirement,[6] we attempted to model these policies by further dividing these eight datasets

[6] As far as we know, so far there has been no such pure (ideal) data publicly available.

based on the minimum length or composition requirement. However, we were cautioned that simply dividing an existing dataset according to some artificial policy may be meaningless, for user behaviors will be largely skewed in this process. A collateral evidence of this is the observation that, passwords created under an explicit policy "cannot be characterized correctly simply by selecting a subset of conforming passwords from a larger corpus" and "such a subset is unlikely to be representative of passwords created under the policy in question" [65]. Fortunately, after careful examination of our eight datasets (see Table 2 and Table 3), we find that:

(1) Only 2.17% passwords in CSDN are shorter than eight characters long, and these short passwords are highly due to the initial loose policy and these remaining 97.83% long passwords due to the later enhanced password policy, and this transition in password policies has been confirmed;

(2) As high as 75.79%(=69.77%+6.02%) passwords in Myspace are composed of both letters and numbers and more than 18.24% users select passwords with a sequence of letters concatenated with the number "1", which highly suggests that there was a transition in composition requirements at sometime before the hacking happened, though by no means can we confirm this transition.

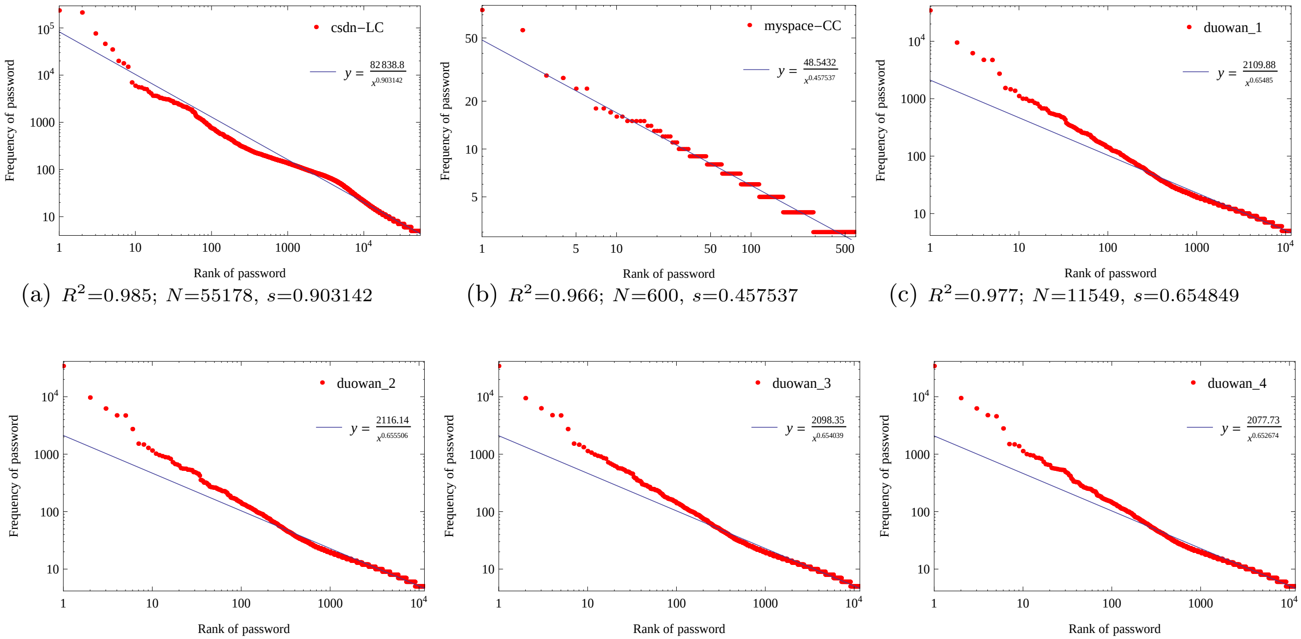
Consequently, these two datasets provide useful subsets that are representative of passwords complying with the above two constrained password policies, respectively. More specifically, 97.83% long passwords from CSDN constitute a dataset created under a policy that requires passwords to be at least eight characters long, and 75.79% passwords from Myspace constitute a dataset created under a policy that requires passwords to be at least one letter and one number. And we call them "csdn-lc" and "myspace-cc" for short, where "lc" stands for "length constrained", and "cc" stands for "character constrained". The linear regression results on these two refined datasets are depicted in Fig.3(a) and 3(b), respectively. We can see that, the coefficient of determination R^2 of these two regressions are 0.966 or higher, indicating a sound fitting. This suggests that Zipf's law can also be applied to passwords created under very constrained policies.

To investigate whether subsets of a dataset that obeys Zipf's law also comply with this law, we further conduct linear regressions on subsets randomly selected from the eight datasets. As expected, there are no significant differences in fitting effect between any of the subsets and their parent dataset (Fisher's exact test, p-valueleq0.001). Due to space constraints, only four randomly selected subsets (each with a size of 1 million) from Duowan are depicted in Fig.3(c) $\sim$ Fig.3(f). As R^2 of these four regressions are all 0.977 and very close to 1, which indicates Zipf's law fits well in these subsets. This implies that if we can obtain a sufficiently large subset of passwords of one website, then the distribution of the whole passwords can be (precisely) determined by conduction a linear regression and fitting to a Zipf's law. Nevertheless, how much fraction of a dataset can be deemed "sufficiently large"? How about one sixth, one tenth, or one hundredth? This suggests a natural direction for future research.

At this stage, a natural question arises: *Can our observations be generalized to all user-generated passwords?* Or equally this question may be expressed as: Whether the datasets used in this work can be representative of all datasets? The answer is highly in the affirmative. On the one hand, the datasets used in this work are so far the most diversified (in terms of application domain, location and language) and among the largest ones (in terms of both the total number of passwords and the number of datasets), and thus they are of sound representativeness. In previous researches on passwords, to the best of our knowledge, the most diversified datasets (i.e., three from US and three from China, and with each from different applications) have been reported in Ma et al.'s work [43] and the largest datasets (i.e., seven datasets with a total of 114 million passwords) have been used in Li et al.'s work [41], while in our work, we employ eight datasets with each from a different application domain and with a total of 56 million passwords. Admittedly, our datasets (as well as the length-constrained and character-constrained ones) cannot represent all sorts of real-life datasets, for instance, none of them represent credentials with great importance (e.g., e-bank accounts). That being said, these datasets still represent a significant number of user-generated passwords and can be applied to investigate the underlying distributions.

On the other hand, rigorously speaking, there is no definite answer to the appropriateness of a generalization like ours. Physicians aim at understanding how the physical world works can never know for sure if their theories (e.g., Newton's laws) are the right ones, instead they can only tell if their theories are consistent with state-of-the-art experiments. Similarly, we aim at understanding how the real-life passwords distribute, but can never know for sure if our theory is definitely correct; With adequate data and right tools, we can only develop theoretical models to characterize the distribution of passwords more and more accurately, and this might be a never-ending work in progress. We freely admit more efforts need to be devoted to this subject.

Overall, although our data is not ideal, we believe that our findings do provide a much better understanding of the distributions of user-generated passwords and can be widely applicable. While so little is known about this important topic, even relatively limited exploration constitutes progress, let alone a fundamental investigation.



(a) R^2=0.985; N=55178, s=0.903142 (b) R^2=0.966; N=600, s=0.457537 (c) R^2=0.977; N=11549, s=0.654849

(d) R^2=0.977; N=11479, s=0.655506 (e) R^2=0.977; N=11571, s=0.654038 (f) R^2=0.977; N=11626, s=0.652674

Fig. 3. Zipf's law in passwords created under constrained policies and in passwords selected randomly from a real-life dataset (using Duowan as a typical example) plotted on a log-log scale

4 Some foundational implications

4.1 Implications for password policies

Recently, many works on password policy (e.g., [15,59]) have suggested to disallow users from choosing dangerously-popular passwords (e.g., `123456` and `password123`) whose occur with probability greater than a predefined threshold $\mathcal{T}$ (e.g., $\mathcal{T} = 1/10,000$). Surprisingly, their motivation is mainly based on the *empirical* observation that some users employ undesirably popular passwords and such passwords are particularly prone to statistic attacks, a form of dictionary attack (maybe either online or offline) in which an attacker sorts her dictionary by popularity and guesses the most popular passwords first, yet so far little rationale has been given and many foundational questions remain to be addressed. For example, what's the fundamental tendency of growth of the fraction of users that will be affected by decreasing the popularity threshold $\mathcal{T}$? What proportion of users will be affected if we restrict the top 0.0001% most popular passwords? How about restrict the top 0.01% most popular passwords?

We are now ready to answer these questions. In the above section, we have shown that in most cases, user-generated passwords perfectly obey Zipf's law, which states that the rank r of a password and its frequency f_r follow the equation $f_r = \frac{C}{r^s}$, where C is a constant that is typically slightly smaller than the frequency of the most popular password (denoted by F_1), i.e., $C = f_1 \leq F_1$. For illustrative purpose, assume the frequency of user password X is a continuous real variable, and the corresponding probability of taking a value in the interval from x to $x + dx$ is denoted by $p(X = x)dx$. According to [2], now $p(X = x)$ obeys a power law distribution. More specifically,

$$p(X = x) = C' \cdot x^{-\alpha} \tag{3}$$

where $\alpha = 1 + 1/s$, s is as defined in Eq.1. As for C', it is given by the normalization requirement that

$$1 = \int_{x_{min}}^{\infty} p(X = x)\, dx = C' \cdot \int_{x_{min}}^{\infty} x^{-\alpha}\, dx = \frac{C'}{1-\alpha}[x^{-\alpha+1}]_{x_{min}}^{\infty} \tag{4}$$

where x_{min}, in practical situations, is defined not to be the smallest value of x measured but to be the smallest for which the power-law behaviour holds. As $\alpha = 1 + 1/s > 1$, we get

$$C' = (\alpha - 1){x_{min}}^{\alpha-1} \tag{5}$$

Consequently, the probability that the frequency of a particular password will be greater than x ($x \geq x_{min}$) is given by

$$P(X > x) = \int_x^\infty p(X = x')\,dx' = \frac{C'}{\alpha - 1}x^{-\alpha+1} = (\frac{x}{x_{min}})^{-\alpha+1} \tag{6}$$

Note that by definition, $P(X > x)$ can also be seen as the cumulative password popularity distribution function. Based on Eq.4 and Eq.5, for $\alpha = 1 + 1/s > 2$ (see Table 5), the least frequency $x_\mathcal{T}$ allowed under a threshold $\mathcal{T}$ can be determined

$$x_\mathcal{T} = \mathcal{T} \cdot \int_{x_{min}}^\infty xp(X = x)\,dx = \mathcal{T} \cdot C' \cdot \int_{x_{min}}^\infty x^{-\alpha+1}\,dx = \mathcal{T} \cdot \frac{C'}{2-\alpha}[x^{-\alpha+2}]_{x_{min}}^\infty = \mathcal{T} \cdot \frac{\alpha-1}{\alpha-2}x_{min} \tag{7}$$

We denote the exact fraction of user accounts (with password frequencies exceeding $x_\mathcal{T}$) that will be *potentially* and *actually* affected by the threshold $\mathcal{T}$ to be $W_p(X > x_\mathcal{T})$ and $W_a(X > x_\mathcal{T})$,[7] respectively, where

$$W_p(X > x_\mathcal{T}) = \frac{\int_{x_\mathcal{T}}^\infty x'p(X = x')\,dx'}{\int_{x_{min}}^\infty x'p(X = x')\,dx'} = (\frac{x_\mathcal{T}}{x_{min}})^{-\alpha+2} \tag{8}$$

$$W_a(X > x_\mathcal{T}) = \frac{\int_{x_\mathcal{T}}^\infty (x' - x_\mathcal{T})p(X = x')\,dx'}{\int_{x_{min}}^\infty x'p(X = x')\,dx'} = \frac{1}{\alpha-1} \cdot (\frac{x_\mathcal{T}}{x_{min}})^{-\alpha+2} \tag{9}$$

Using Eqs.6∼9, we can get the fraction of user accounts with each of its password lies in the most popular part $P(X > x_\mathcal{T})$:

$$W_p(X > x_\mathcal{T}) = (P(X > x_\mathcal{T}))^{(-\alpha+2)/(-\alpha+1)} \tag{10}$$

Since $\alpha = 1 + 1/s$, Eq.10 can be re-written as

$$W_p(X > x_\mathcal{T}) = (P(X > x_\mathcal{T}))^{(1-\frac{1}{s})/(-\frac{1}{s})} = (P(X > x_\mathcal{T}))^{1-s} \tag{11}$$

Similarly, Eq.9 can be re-written as

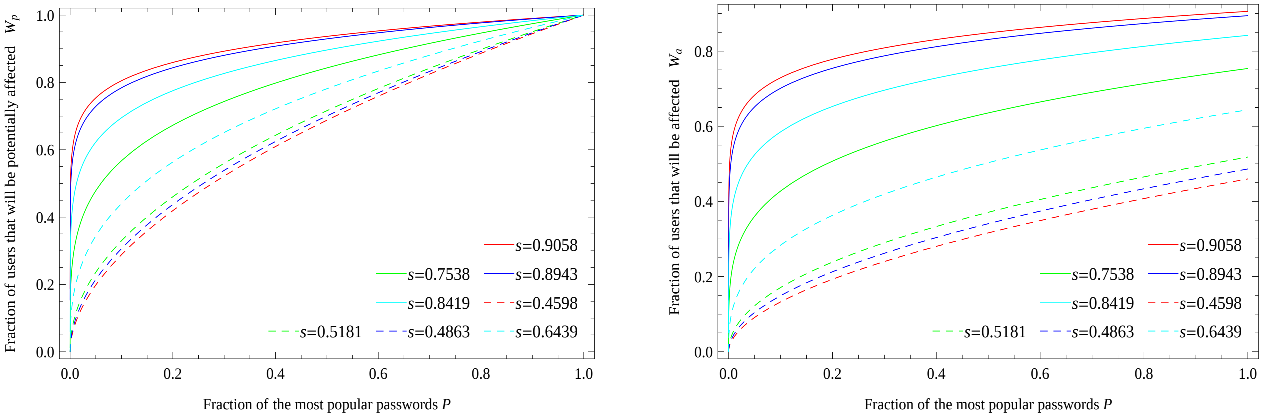
$$W_a(X > x_\mathcal{T}) = \frac{1}{\alpha-1} \cdot (P(X > x_\mathcal{T}))^{(1-\frac{1}{s})/(-\frac{1}{s})} = s \cdot (P(X > x_\mathcal{T}))^{1-s} \tag{12}$$

This suggests that the two reduced-usability indicators $W_p(X > x_\mathcal{T})$ and $W_a(X > x_\mathcal{T})$ follow a Pareto's law with a positive exponent $1 - s$ regarding the cumulative password popularity distribution function $P(X > x_\mathcal{T})$. For an intuitive grasp, in Fig.4 we depict the form of the curves of $W_p(X > x_\mathcal{T})$ and $W_a(X > x_\mathcal{T})$ against $P(X > x_\mathcal{T})$ for various values of s as listed in Table 5.

The steep increase of W_p and W_a at the very beginning of their curves (see Fig.4) explicitly reveal that, popular passwords are overly popular and a non-negligible fraction of users will be annoyed even if an extremely small proportion of popular passwords are checked. For example, according to Eq.12, W_a=2.51% users will be annoyed when $s = 0.7538$, $\mathcal{T} = 1/1024$ and $P = 0.0001\%$. To see whether our theory accords with the reality, we also summarize the statistical results from eight real-life password datasets in Table 7. One can confirm that, the theoretical W_a exceeds the empirical W_a by a factor of $1 \sim 3$. The main reason why the results obtained from our theoretical model are larger than the experimental statistical results is that, there is a large proportion of passwords that are not frequent (i.e., their frequencies are below x_{min}), which is generally called the "noisy tail" [50] in the statistical domain. In addition, for simplicity we have modelled the frequency of a user password, which is a discrete integer, to be a continuous real variable, this will inevitably introduce some deviations.

Though the above theoretical model is not exactly accurate for real-life passwords, it does reveal the fundamental tendency of the fraction of users that will be affected by a popularity threshold and provides insightful, concise and practical indicators that facilitate policy designers and system administrators to offer a more acceptable trade-off between usability and security. For example, under our theory it is unreasonable set $\mathcal{T} = 1/1000,000$ for Internet-scale websites (let alone small sites), for more than 60% users will be potentially annoyed. Unfortunately, the policy in [59] just suggested such a value for $\mathcal{T}$. On the other hand, the Zipf's law identified in Sec.3.4 suggests that the frequencies of passwords descend at a approximately logarithmic rate, and thus only a limited proportion of

[7] Note that, $W_p(X > x_\mathcal{T})$ and $W_a(X > x_\mathcal{T})$ are indeed two independent and useful indicators to measure the extent to which usability will be affected. For instance, now if www.dodonew.com enforces a popularity-based password policy with $\mathcal{T} = 1/1024$, then there will be $W_p(X > x_\mathcal{T})$=3.33% accounts with passwords more popular than $\mathcal{T} = 1/1024$, which means each of these 3.33% accounts has an equal potential to be required to change a new password. However, there will only be $W_a(X > x_\mathcal{T})$=2.51% accounts that will actually be required to choose a different password for the reason that, after $W_a(X > x_\mathcal{T})$=2.51% accounts have already been changed, the remaining $W_p(X > x_\mathcal{T}) - W_a(X > x_\mathcal{T})$=0.82% accounts will be with passwords less popular than $\mathcal{T} = 1/1024$.



(a) Fraction of users that will be *potentially* affected with $P(X > x_{\mathcal{T}})$ (b) Fraction of users that will be *actually* affected with $P(X > x_{\mathcal{T}})$

Fig. 4. The fraction of users that will be potentially/actually affected by a popularity-based policy, if passwords is distributed following a Zipf law with exponent s as listed in Table 5.

Table 7. Effects of password popularity threshold $\mathcal{T}$ on the fraction of passwords with undesirable popularity (i.e., P) and on the fraction of user accounts that will be actually affected (i.e., W_a)

Password Dataset	$\mathcal{T}$ =1/1024		$\mathcal{T}$ =1/10000		$\mathcal{T}$ =1/16384		$\mathcal{T}$ =1/100000		$\mathcal{T}$ =1/1000000	
	P	W_a	P	W_a	P	W_a	P	W_a	P	W_a
Tianya	0.0001%	6.6023%	0.0015%	10.7586%	0.0023%	11.6473%	0.0394%	16.9065%	0.4416%	30.9110%
Dodonew	0.0001%	1.3926%	0.0009%	3.1556%	0.0014%	3.6298%	0.0132%	5.7710%	0.2958%	11.2351%
CSDN	0.0002%	9.4648%	0.0029%	12.2806%	0.0049%	12.8732%	0.0912%	16.6093%	0.8441%	24.6874%
Duowan	0.0004%	5.8130%	0.0048%	8.8648%	0.0079%	9.6064%	0.1032%	13.0349%	1.6607%	24.4955%
Myspace	0.0054%	0.1228%	0.5358%	2.1952%	1.9007%	4.6961%	–	–	–	–
Singles.org	0.1553%	2.6154%	14.1818%	24.7138%	–	–	–	–	–	–
Faithwriters	0.1917%	1.3390%	–	–	–	–	–	–	–	–
Hak5	3.2327%	10.3113%	–	–	–	–	–	–	–	–

Note: "–" stands for "not applicable", due to the mere fact that $1/\mathcal{T}$ is larger than the size of corresponding dataset.

passwords are overly popular. Consequently, we only need to prevent these overly passwords and set an appropriate popularity threshold $\mathcal{T}$. For instance, less than 13% users for most sites will be annoyed when $\mathcal{T}$ is set to the moderate value 1/16384 complying with a Level 2 certification [13], which suggests that $\mathcal{T} = 1/16384$ is acceptable for many Internet-scale e-commerce websites. This for the first time provides a sound rationale (foundation) that explicates the necessity and feasibility (as well as precautions) for popularity-based password policies.

4.2 Implications for password-based authentication

Another foundational implication is for provably secure authentication protocols that involve passwords, like password-only single-factor schemes (e.g., two-party [35] and multi-party [16]) and password-based multi-factor schemes (e.g., two-factor [70] and three-factor [30]). Here we first show the implication for password-only schemes, which are also called PAKE protocols. In most PAKE protocols with provable security (e.g., [7,16,55] in the random oracle model and [27,35,36] in the standard model), it is typically assumed that "password pw_C (for each client C) is chosen independently and uniformly at random from a dictionary $\mathcal{D}$ of size $|\mathcal{D}|$, where $|\mathcal{D}|$ is a fixed constant which is independent of the security parameter k", then a security model is described, and finally a "standard" definition of security as the one in [35] is given:

> "$\cdots$ $\cdots$ Protocol $\mathcal{P}$ is a secure protocol for password-only authenticated key-exchange if, for all [password] dictionary sizes $|\mathcal{D}|$ and for all ppt[probabilistic polynomial time] adversaries $\mathcal{A}$ making at most $Q(k)$ on-line attacks, there exists a negligible function $\epsilon(\cdot)$ such that:
>
> $$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/|\mathcal{D}| + \epsilon(k) \tag{13}$$
>
> where $\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k)$ is the advantage of $\mathcal{A}$ in attacking $\mathcal{P}$."[8]

[8] We remark that some PAKE protocols (e.g., [7,16]) relax this definition of security to $\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq c \cdot Q(k)/|\mathcal{D}| + \epsilon(k)$, where c is constant positive integer, which indicates the adversary now is allowed to guess c passwords per on-line attempt. However, this does not necessarily mean that the corresponding protocol is actually subject to the threat that an adversary can guess more than one passwords per on-line attempt, for this relaxation may be due to some technical reasons in the reductionist proofs but not an inherent protocol vulnerability.

According to Bonneau's work [10], generally user-generated passwords offer about $20 \sim 21$ bits of actual security against an optimal offline dictionary attack, i.e., the practical value of $|\mathcal{D}|$ is about $2^{20} \sim 2^{21}$. This indicates that a system which employs a PAKE protocol achieving the security goal of Eq.13 can assure that one online guessing attempt will attain a success rate no larger than $1/2^{20} \sim 1/2^{21}$, which is apparently not the case in practice.

As a prudent side note, some of these works (e.g., $[27, 35, 36]$) complement that the assumption of a uniform distribution of passwords with a constant-size dictionary is made for simplicity only, and their security proofs can be extended to handle more complex cases where passwords do not distribute uniformly, or different distributions for different clients. However, such a complement only serves to obscure their security statements and undermine the readers' (e.g., people in industry, government, and academia) understanding of to exactly what extent they can have confidence in the authentication protocol used to protect systems, for no one knows what's the distribution then if "passwords do not distribute uniformly". This defeats the purpose of constructing provably secure protocols which "explicitly capture the inherently quantitative nature of security, via a concrete or exact treatment of security" and "offer quantitative security guarantee" [6] in the first place.

According to our theory, now it is fundamentally unnecessary (unrealistic) to make an assumption of uniform distribution of passwords, instead one may directly make the Zipf assumption about password distributions. Since system assigned random passwords $[58, 60]$ is hardly usable, most systems allow users to generate their own passwords, which would highly lead to the passwords complying with the Zipf distribution as we have shown in the previous section. However, under the Zipf assumption, it is highly likely that $\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = \frac{C/1^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \frac{C/2^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \cdots + \frac{C/k^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} = \frac{\sum_{j=1}^{k}\frac{1}{j^s}}{\sum_{i=1}^{|\mathcal{D}|}\frac{1}{i^s}}$ will be alarmingly large (due to the approaching-logarithm-function nature of harmonic numbers [54]) and the system is in serious danger, even if k is rather small. For instance, this value reaches 1.49% when making only 3 on-line attacks for the website www.dodonew.com which enables monetary transactions. As can be seen in Table 4, seven of eight websites have a chance of more than 3.28% being breached by an attacker who makes merely ten online impersonation attempts.

From Eq.12 (or Fig.4(a)) we can also see that, a rather small fraction of the most popular passwords (denote by P) can account for a non-negligible proportion of user accounts (denote by W_a), which suggests that an online guessing attacker can succeed with a chance W_a just by trying $P \cdot |\mathcal{D}|$ different passwords. In other words, even if the authentication protocol employed is provably secure, secure user identification still cannot be reached if the passwords of the system obey Zipf's law. This once again highlights that cryptographic methods should be compounded with systematic solutions to assure system security. To this end, the passwords shall not follow a Zipf distribution. This indicates that some necessary countermeasures (e.g., exploiting some password policies that restrict the overly popular passwords) shall be taken, which may lead to a set of passwords with *a skewed Zipf distribution*.

While the uniform distribution assumption made about passwords is unrealistic, the Zipf distribution is insecure and the skewed Zipf distribution seems hardly possible to be rigorously characterized, we are stuck in a conundrum to formulate the definition of security results like Eq.13. Inspired by the essential notion of security that a secure PAKE protocol can provide – only oneline impersonation attacks are helpful to the adversary in breaking the security of the protocol [27], we manage to get out of the problem by giving up the idea of firstly characterizing the distribution of password and then formulate the definition of security and instead, we provide a tight upper bound for the adversary's advantage. More specifically, Eq.13 now is improved as follows:

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq F_1 \cdot Q(k)/|\mathcal{DS}| + \epsilon(k) \tag{14}$$

where F_1, as said earlier, denotes the frequency of the most popular password in the password dataset $\mathcal{DS}$ and the other notations are of the same meaning with those of Eq.13. Note that, dictionary $\mathcal{D}$ is the password sample space and it is a *set*, while dataset $\mathcal{DS}$ is a (specific) password sample and it is a *multiset*. Therefore, the value of $F_1/|\mathcal{DS}|$ is *exactly the threshold probability* $\mathcal{T}$ (e.g., $\mathcal{T}$ =1/10,000) that the underlying password policy maintains. For a system to reach a Level 1 certification [13], the success chance of an online guessing attacker should be no larger than 1 in 1024, which indicates $F_1/|\mathcal{DS}| \leq 1/1024$; Similarly, a Level 2 certification, the system shall ensure $F_1/|\mathcal{DS}| \leq 1/16384$. For example, for the gaming and e-commerce website www.dodonew.com to achieve a Level 2 security, F_1 should have been no larger than $991(\approx 16231271/16384)$.

Also note that, Eq.13 is actually a special case of Eq.14, where $F_1 = 1$ and $|\mathcal{DS}| = |\mathcal{D}|$. Moreover, Eq.14 can be also used to roughly formulate the formal security results in situations where user passwords perfectly follow Zipf's law. Though not precisely accurate, Eq.14 (e.g., let $F_1/|\mathcal{DS}| = 1/100$) is still more reasonable (and realistic) than the ones like Eq.13 that are currently widely used in the cryptographic protocol community (e.g., $[5, 35, 55]$).

We happen to find that a very recent PAKE protocol (named SPOKE) proposed by Abdalla et al. [1] use a different formulation of security from traditional ones:

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/2^m + \epsilon(k) \tag{15}$$

where m is the *min-entropy* of the passwords.[9] Actually, it is not difficult to see that this kind of formulation (i.e., Eq.15) is in essential the same with our Eq.14, for one can derive that $m = -\log_2(F_1/|\mathcal{DS}|)$ [10]. By contrast, our formulation is much more concrete and intelligible. What's more, no rationale or justification for preferring Eq.15 but not Eq.13 is given in [1] . Fortunately, in the above section we have at last settled this fundamental issue: Are passwords likely to be of a uniform distribution or not? And, if not, then what should be the right distribution?

One can also see that, if m is defined to be the *entropy* of the passwords, then Eq.15 is virtually equal to Eq.13 and it provides a *mean* value for the online guessing difficulty, for one can derive that $m = \sum_{r=1}^{|\mathcal{D}|} -p_i \log_2 p_i$, where p_i is the probability of the i^{th} most frequent password in $\mathcal{D}$ (e.g., $p_1 = F_1/|\mathcal{DS}|$). This well explicates why Benhamouda et al. (see Sec.6.1 of [8]) state that "equivalently the advantage of any adversary can be bounded" by either Eq.13 or Eq.15. However, if m is defined to be the *min-entropy* of the passwords, which is the right definition in most cases, we have shown that Eq.13 and Eq.15 (or equally, Eq.14) will be significantly different from each other.

Unlike PAKE protocols where users have to interact with the server to register their passwords, most multi-factor schemes (e.g., [63,70,75]) provide a property, which is termed "DA2-Local-Secure" [69], to facilitate users change their passwords freely and locally (i.e., without any interaction with the remote server). Since there is no interaction with the remote server, popularity-based password policy cannot be enforced, user passwords will *almost definitely* follow a Zipf distribution. However, when evaluating whether "truly multi-factor security" can be provided, these schemes typically perform a reductionist security proof and obtain a security result like Eq.13 (see definition 1 of [75]), under the assumption that the other factor(s) except the password factor has been compromised. As discussed above, our theory discourages such simple but unrealistic, actually misleading (i.e., a false sense of security) form of formulation. A formulation like our proposed Eq.14 is more accurate and appropriate for such cases. This further suggests the necessity of abandoning the property "DA2-Local-Secure" and requiring users to change their passwords by interacting with the server (i.e., preferring the property "DA2-Interactive" [69]), providing an answer to the open problem raised in [69]: As an ideal scheme that achieves all the criteria (including ten desirable properties and nine security goals) is beyond attainment, then which criterion should be abandoned?

To the best of knowledge, we for the first time pay attention to the joint between passwords and password-based single-factor or multi-factor authentication protocols. With the knowledge of the exact distribution of passwords, we manage to develop a more accurate, realistic and versatile formulation to characterize the formal security result for password-based authentication protocols. Here we take password-based authentication as a case study, one can easily find that our results also can be readily applied to other kinds of password-based cryptographic protocols such as password-protected secret sharing (e.g., [5]) and password-based signatures (e.g., [26]).

5 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.

5.1 Our metric

Normally, the offline guessing attacker,[10] 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 the target password is matched. In the extreme case, if the attacker has also obtained the entire password dataset in plain-text and thus, she can obtain the right order of the passwords, this attack is called an optimal attack [10, 20].[11] Accordingly, we can use the cracking result $\lambda^*(n)$ to be the strength metric of a given password dataset:

$$\lambda^*(n) = \frac{1}{sum} \sum_{r=1}^{n} f_r \tag{16}$$

[9] We note that, in Sec.3.2 of [1], m is re-defined to be the *entropy* of the passwords. This inconsistence would lead to great differences in security notions. We conjecture a typo has occurred here.

[10] The attacks mentioned in this Section are all offline guessing attacks, for our purpose is to measure the strength of an entire dataset, which is generally characterized by how much percentage of passwords could be successfully covered (guessed).

[11] Note that, the optimal attack is of theoretic value (i.e., the upper bound) to characterize the best attacking strategy that an attacker can adopt. In practice, if an attacker has already obtained all the plain-text passwords, there is no need for her to order these passwords to crack themselves.

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):

$$\lambda^*(n) \approx \lambda(n) = \frac{\sum_{r=1}^{n} \frac{1}{r^s}}{\sum_{r=1}^{N} \frac{1}{r^s}} \tag{17}$$

It should be noted that, in Eq.17, $\lambda^*(n)$ 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.16 and $\lambda(n)$ as a function of n according to Eq.17, and put these two curves together to see how they agree with each other. In Fig.5, 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.3.4, here we do not illustrate the related picture for Hak5 dataset. As for the other six datasets, see the miniatures in Fig.6.

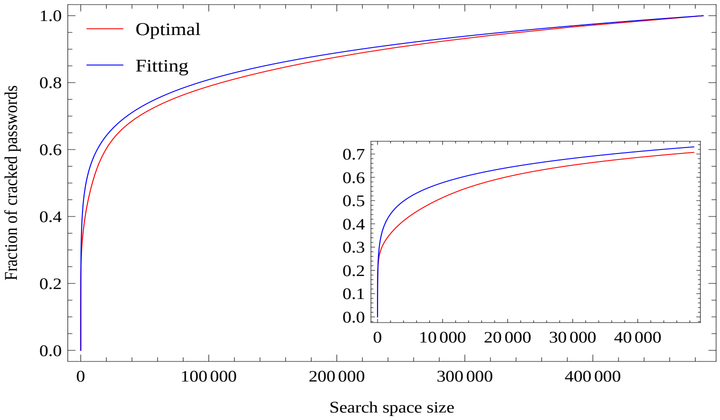
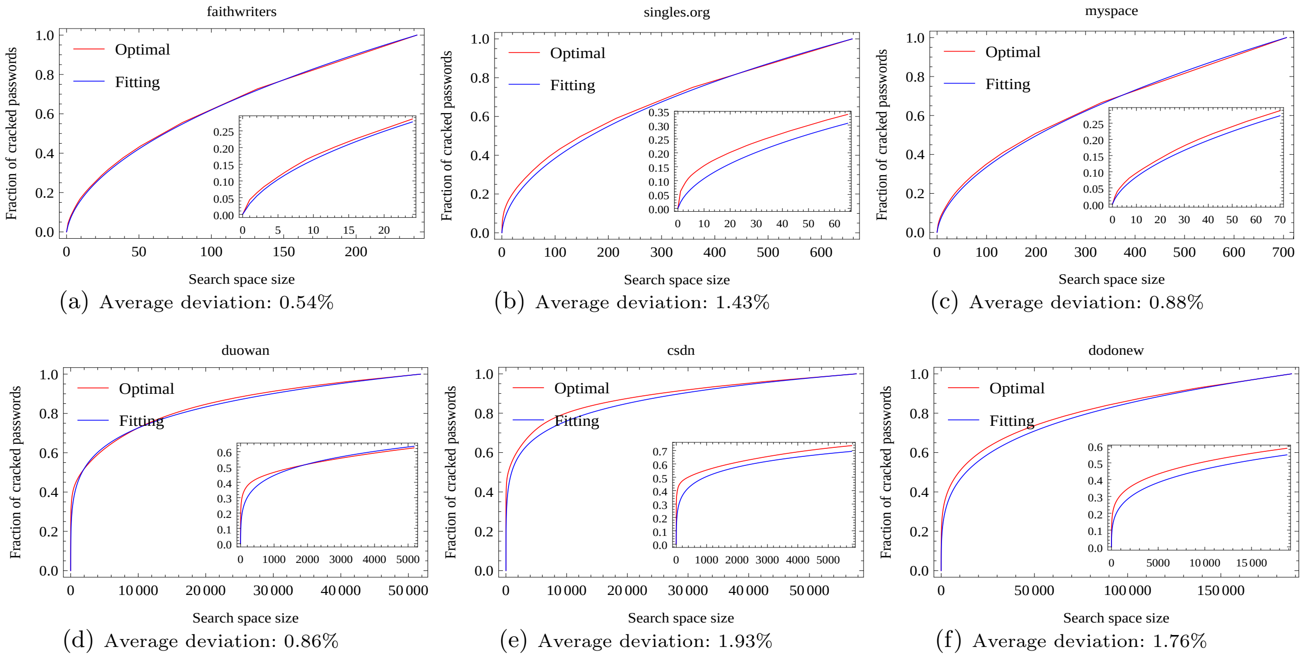


Fig. 5. Consistence of optimal attack with our fitting metric on Tianya (Average deviation is 1.32%)



(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. 6. Consistence of effectiveness between real optimal attack and our fitting metric

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 [34] 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*

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

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.5.2, and in Section 5.3 its compliance with cracking results will be shown by the optimal attack and the state-of-the-art cracking algorithm (i.e., PCFG [72]), 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 [72] or markov-based [49]) 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.

Some Remarks. Note that, as with the entropy metric recommended in the NIST SP800-63-2 document [13] and the α-guesswork proposed in [10], our metric is not effective on password datasets that are in clear-text or un-salted hash. This is an inherent limitation of all statistic-based metrics (e.g., [10, 13] and ours). For salted-hash passwords, one needs to resort to attacking-based approaches (e.g., [37, 43]), albeit at the cost of reduced accuracy (as we will show in Sec.5.3, attacking-based approaches in their current form have too many uncertainties and are far from accurate). It is also worth noting that, there could be weak policies that result in a good metric, like requiring users to type their usernames as the start of a password. Obviously, this would make all passwords more unique and leads to a better metric, but it wouldn't at all increase the resistance of passwords if the attacker knows the underlying policy. This constitutes another limitation of statistic-based metrics. In this case, one also needs to resort to attacking-based approaches.

Nevertheless, these and other limitations does not affect much the applicability of our metric mainly for several reasons. Firstly, our metric can rely on *a subset* of the entire dataset and only involves *offline operations* to be performed after a relatively long period of time (e.g., a year), and thus the website can implement salted passwords, which are online, to authenticate users and maintain a subset of passwords in un-salted hash, which are physically offline and well protected, to facilitate our measurement. Secondly, 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 [56]) are still in un-salted form. Now, it is time for these legacy sites to take actions, an important part of which is to access its password policy. And our metric is the right choice. Thirdly, it is well known that the authorities in many countries (e.g., The National Security Agency of U.S. [47]) 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. Last but not the least, even if no plain-text (or unsalted-hash) passwords from real-life websites are available, field experiments (e.g., [23,39]) can be used to collect user generated passwords. With these field passwords, our metric can be used to help password policy designers and system administrators assess the goodness of a given password policy in terms of security before it is put into any practical use.

In a nutshell, despite its limitations, our metric is realistically practical in many realistic scenarios, and it can be complemented by attacking-based approaches in cases where statistic-based approaches are not applicable. As we will show in the following, our metric also develops the state-of-the-art statistic-based approach proposed by Bonneau [10].

5.2 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 $1 \leq n \leq N_B$, we have

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

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

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

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 $1 \leq n \leq N - 1$,

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

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

$$\lambda_A(n) < \lambda_C(n) \quad 1 \leq n \leq N_A - 1$$

$$\lambda_C(n) < \lambda_B(n) \quad 1 \leq n \leq N_A - 1$$

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

Interestingly, we observe that, based on the conditions of Theorem 1, the three statistical-based metrics (i.e., λ_β, μ_α, G_α) for assessing the strength of a given password dataset proposed by Bonneau [10] 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 [10] for more details.

It is not difficult to see that λ_β is essentially the $\lambda^*(n)$ as in Eq.16, where β is analogous to n. According to Eq.17 and our Theorem 1, we get $\mu_\alpha(A) \geq \mu_\alpha(B)$. 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 $0 < \alpha \leq 1$.

5.3 Experimental results

In this subsection, we further use the simulated optimal attack and the state-of-the-art password attacking algorithm (i.e., PCFG [37,72]) on real-life password datasets to demonstrate that our metric in Sec.5.1 is practically effective.

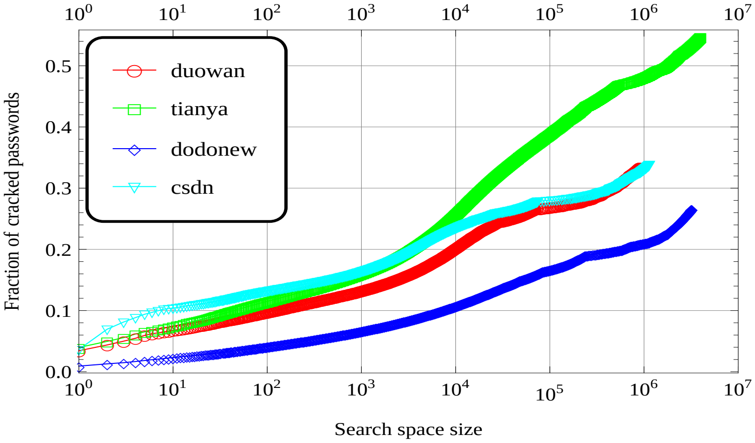
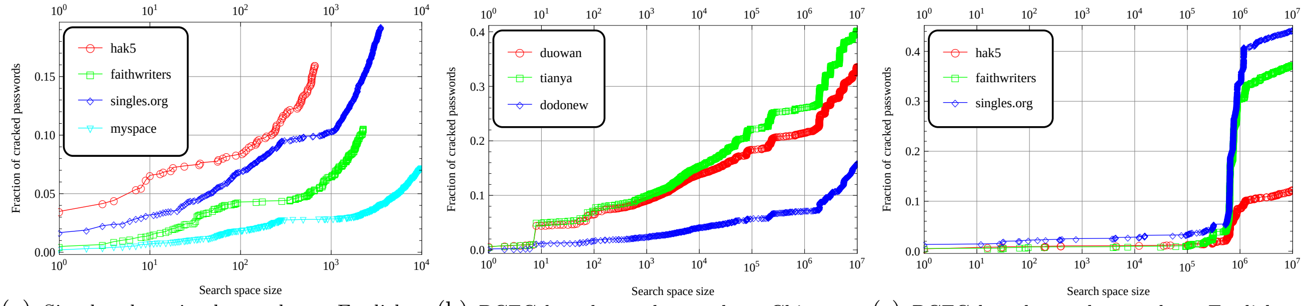


Fig. 7. Simulated optimal attack on four Chinese datasets (i.e., Tianya, Dodonew, Duowan and CSDN)

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



(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. 8. Simulated optimal attacks and PCFG-based attacks on different groups of datasets

optimal attack". For better presentation, we divide the eight datasets into two groups: Group one with dataset sizes all larger than one million and group two smaller than one million. Simulated optimal attacking results on group one are illustrated in Fig.7, and results on group two are illustrated in Fig.8(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 5. For instance, from Fig.7 we know that, for any search space size (i.e., ever n), dataset Tianya is weaker than dataset Duowan, which implies $N_{\text{tianya}} > N_{\text{duowan}}, s_{\text{tianya}} < s_{\text{duowan}}$. This implication accords with the statistics in Table 5.

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 [21], "SogouLabDic.dic"[12] and the 20 million hotel reservations dataset[13]) as the input dictionary. The results are depicted in Fig.8(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" [53] and "english_lower.lst" from [21]) as the input dictionary. The results are depicted in Fig.8(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 5, 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.8(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, for 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., [37, 71]) 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.

[12] `http://www.sogou.com/labs/dl/w.html`

[13] `http://www.4hoteliers.com/news/story/12047`

6 Conclusion

In this work, we have provided a novel prospective of the distributions of real-life (user-generated) passwords. By adopting techniques from computational statistics, we for the first time show that Zipf's law describes concisely skewed distributions of passwords. We have further investigated the general applicability of our observations and discussed multiple benefits from understanding the distribution of passwords. Particularly, most of the existing password-based protocols (e.g., [7, 16, 27, 35, 55]) have been proven secure under the hypothesis that passwords are uniformly distributed, yet we show that their formulations of security results fails to capture the realistic distribution of real-life passwords and may have some unintended consequences. Accordingly, we suggest a new formation to more accurately characterize the formal security results.

As one specific application of our Zipf theory, a new statistic-based metric that measure the strength of password datasets is put forward. Our metric outperforms most of the existing statistic-based metrics in accuracy (e.g., [13]) or simplicity (e.g., [10]). Of great interest is its deterministic nature of measurement of the password dataset strength, which facilitates system administrators to conduct fair and precise comparisons among different datasets. We have formally proved our metric in a mathematically rigorous manner and also revealed some implications from Bonneau's α-guesswork [10]. We have 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 remain to be investigated. For example, do six-digit PINs obey Zipf's law? Do extremely high value accounts (e.g., e-banking passwords) obey Zipf's law? How will the password distributions of a system change as time goes on? It is a mixed blessing that the opportunities for such investigations to be conducted in the future are only growing as more sites of high values are compromised and more datasets are made available. There are also many new issues brought about by the findings in this work. For example, as our findings suggest the necessity of employing some password creation policy like Schechter et al. [59] which only allows passwords with popularity lower than a threshold, how should we set this threshold? And to what extent usability will be affected? Is it necessary for multi-factor authentication protocols to give up the feature of supporting users in changing their passwords without interaction with the remote server? We believe this paper will trigger discussions about the fundamental implications that progresses in the passwords analytics (e.g., password cracking, usability and policy) would have for the area of password-based cryptographic protocols (e.g., authentication, signature and secret sharing).

References

1. Abdalla, M., Benhamouda, F., Pointcheval, D.: SPOKE: Simple password-only key exchange in the standard model. Cryptology ePrint Archive, Report 2014/609 (2014), https://eprint.iacr.org/2014/609.pdf
2. Adamic, L.A.: Zipf, Power-laws, and Pareto - a ranking tutorial (Mar 2014), http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html
3. 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)
4. Axtell, R.L.: Zipf distribution of US firm sizes. Science 293(5536), 1818–1820 (2001)
5. Bagherzandi, A., Jarecki, S., Saxena, N., Lu, Y.: Password-protected secret sharing. In: Proc. CCS 2011. pp. 433–444. ACM (2011)
6. Bellare, M.: Practice-oriented provable-security. In: Damgard, I. (ed.) ISC 1999, LNCS, vol. 1561, pp. 1–15. Springer Berlin/Heidelberg (1999)
7. 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)
8. Benhamouda, F., Blazy, O., Chevalier, C., Pointcheval, D., Vergnaud, D.: New techniques for sphfs and efficient one-round pake protocols. In: Canetti, R., Garay, J. (eds.) CRYPTO 2013, LNCS, vol. 8042, pp. 449–475. Springer Berlin/Heidelberg (2013)
9. Bishop, M., V Klein, D.: Improving system security via proactive password checking. Computers & Security 14(3), 233–249 (1995)
10. 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)
11. 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)
12. Bowes, R.: Password dictionaries (Oct 2011), https://wiki.skullsecurity.org/Passwords
13. 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)

14. 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)
15. Castelluccia, C., Dürmuth, M., Perito, D.: Adaptive password-strength meters from markov models. In: Proc. NDSS 2012. pp. 1–15 (2012)
16. Chen, L., Lim, H.W., Yang, G.: Cross-domain password-based authenticated key exchange revisited. ACM Trans. Inform. Syst. Secur. 16(4), 15 (2014)
17. 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)
18. Constantin, L.: Security Gurus 0wned by Black Hats (July 2009), http://news.softpedia.com/news/Security-Gurus-0wned-by-Black-Hats-117934.shtml
19. Davies, C., Ganesan, R.: Bapasswd: A new proactive password checker. In: 16th National Computer Security Conference. pp. 1–15 (1993)
20. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: Proc. INFOCOM 2010. pp. 1–9. IEEE (2010)
21. Designer, S.: John the Ripper password cracker (Feb 1996), `http://www.openwall.com/john/`
22. Dürmuth, M.: Useful password hashing: how to waste computing cycles with style. In: Proc. NSPW 2013. pp. 31–40. ACM (2013)
23. Egelman, S., Sotirakopoulos, A., Muslukhov, I., Beznosov, K., Herley, C.: Does my password go up to eleven?: the impact of password meters on password selection. In: Proc. of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2013). pp. 2379–2388. ACM (2013)
24. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: Proc. SIGCOMM 1999. pp. 251–262. ACM, New York, NY, USA (1999)
25. Furusawa, C., Kaneko, K.: Zipfs law in gene expression. Physical review letters 90(8), 088102 (2003)
26. Gjosteen, K., Thuen, O.: Password-based signatures. In: Petkova-Nikova, S., Pashalidis, A., Pernul, G. (eds.) EuroPKI 2011, LNCS, vol. 7163, pp. 17–33. Springer Berlin Heidelberg (2012)
27. Halevi, S., Krawczyk, H.: Public-key cryptography and password protocols. ACM Trans. Inform. Syst. Secur. 2(3), 230–268 (1999)
28. Herley, C., Van Oorschot, P.: A research agenda acknowledging the persistence of passwords. IEEE Security & Privacy 10(1), 28–36 (2012)
29. Houshmand, S., Aggarwal, S.: Building better passwords using probabilistic techniques. In: Proc. ACSAC 2012. pp. 109–118. ACM (2012)
30. Huang, X., Xiang, Y., Chonka, A., Zhou, J., Deng, R.H.: A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. IEEE Trans. Parallel Distrib. Syst. 22(8), 1390–1397 (2011)
31. 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)
32. Jakobsson, M., Dhiman, M.: The benefits of understanding passwords. In: Proc. HotSec 2012. pp. 1–6. USENIX Association (2012)
33. Johnston, C.: Why your password cant have symbols (April 2013), `http://arstechnica.com/security/2013/04/why-your-password-cant-have-symbols-or-be-longer-than-16-characters/`
34. 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/
35. Katz, J., Ostrovsky, R., Yung, M.: Efficient and secure authenticated key exchange using weak passwords. J. ACM 57(1), 1–41 (2009)
36. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. Journal of Cryptology 26(4), 714–743 (2013)
37. 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)
38. 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)
39. Komanduri, S., Shay, R., Kelley, P.G., Mazurek, M.L., Bauer, L., Christin, N., Cranor, L.F., Egelman, S.: Of passwords and people: measuring the effect of password-composition policies. In: Proc. of CHI 2011. pp. 2595–2604. ACM (2011)
40. Kuo, C., Romanosky, S., Cranor, L.F.: Human selection of mnemonic phrase-based passwords. In: Proc. SOUPS 2006. pp. 67–78. ACM (2006)
41. Li, Z., Han, W., Xu, W.: A large-scale empirical analysis on chinese web passwords. In: Proc. USENIX Security 2014. pp. 1–16 (Aug 2014)
42. Long, J.: No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing. Syngress (2011)
43. 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)
44. 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)

45. Malone, D., Maher, K.: Investigating the distribution of password choices. In: Proc. WWW 2012. pp. 301–310. ACM (2012)
46. Martin, R.: Amid Widespread Data Breaches in China (Jan 2012), https://sg.finance.yahoo.com/news/Amid-Widespread-Data-Breaches-pennolson-706259476.html
47. 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
48. Morris, R., Thompson, K.: Password security: A case history. Communications of the ACM 22(11), 594–597 (1979)
49. Narayanan, A., Shmatikov, V.: Fast dictionary attacks on passwords using time-space tradeoff. In: Proc. CCS 2005. pp. 364–372. ACM (2005)
50. Newman, M.: Power laws, pareto distributions and zipf's law. Contemporary Physics 46(5), 323–351 (2005)
51. Nievergelt, Y.: Total least squares: State-of-the-art regression in numerical analysis. SIAM review 36(2), 258–264 (1994)
52. Oechslin, P.: Making a faster cryptanalytic time-memory trade-off. In: Proc. CRYPTO 2003. pp. 617–630. Springer (2003)
53. Outpost9.com's Lab: Word lists (Feb 2014), `http://www.outpost9.com/files/WordLists.html`
54. Paule, P., Schneider, C.: Computer proofs of a new family of harmonic number identities. Advances in Applied Mathematics 31(2), 359–378 (2003)
55. Pointcheval, D.: Password-based authenticated key exchange. In: Fischlin, M., Buchmann, J., Manulis, M. (eds.) Proc. PKC 2012, LNCS, vol. 7293, pp. 390–397. Springer Berlin Heidelberg (2012)
56. Prerad, M.: 20 Biggest Data Breaches of 2013, https://www.linkedin.com/today/post/article/20140224 081155-67886711-20-biggest-data-breaches-of-2013
57. Rao, A., Jha, B., Kini, G.: Effect of grammar on security of long passwords. In: Proc. of CODASPY 2013. pp. 317–324. ACM (2013)
58. Scarfone, K., Souppaya, M.: Nist sp800-118: Guide to enterprise password management. Tech. rep., NIST, Reston, VA (April 2009)
59. 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)
60. Shay, R., Kelley, P.G., Komanduri, S., Mazurek, M.L., Ur, B., Vidas, T., Bauer, L., Christin, N., Cranor, L.F.: Correct horse battery staple: Exploring the usability of system-assigned passphrases. In: Proc. SOUPS 2012. pp. 1–20. ACM (2012)
61. Spafford, E.: Observations on reusable password choices. In: Proc. USENIX Security Workshop (1992)
62. Spafford, E.H.: Opus: Preventing weak password choices. Computers & Security 11(3), 273–278 (1992)
63. Tsai, J.L., Lo, N.W., Wu, T.C.: Novel anonymous authentication scheme using smart cards. IEEE Trans. Ind. Inform. 9(4), 2004–2013 (2013)
64. Turan, M.S., Barker, E., Burr, W., Chen, L.: Nist sp800-132: Recommendation for password-based key derivation. Tech. rep., NIST, Reston, VA (Dec 2010), `http://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf`
65. 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)
66. Verheul, E.R.: Selecting secure passwords. In: Abe, M. (ed.) Proc. CT-RSA 2007, LNCS, vol. 4377, pp. 49–66. Springer Berlin / Heidelberg (2007)
67. Vincent, J.: A survey of 2013's most popular passwords (Jan 2014), `http://splashdata.com/press/worstpasswords2013.htm`
68. Wang, D., Chen, H.: Effects of sample size and least frequency on linear regression when simulating a given zipf distribution (Aug 2014), `http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/simulated_zipf.pdf`
69. Wang, D., He, D., Wang, P., Chu, C.H.: Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. IEEE Trans. on Dependable and Secure Computing (2014), `http://dx.doi.org/10.1109/TDSC.2014.2355850`
70. Wang, D., Wang, P., Ma, C.G., Chen, Z.: iPass: Robust smart card based password authentication scheme against smart card loss problem. J. Comput. Syst. Sci. (2014), full version `http://eprint.iacr.org/2012/439.pdf`
71. 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)
72. 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)
73. Wu, T.: A real-world analysis of kerberos password security. In: Proc. NDSS 1999. pp. 13–22 (1999)
74. 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)
75. Yang, G., Wong, D., Wang, H., Deng, X.: Two-factor mutual authentication based on smart cards and passwords. J. Comput. Syst. Sci. 74(7), 1160–1172 (2008)
76. 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)
77. Zipf, G.K.: Human behavior and the principle of least effort. Addison-Wesley Press (1949)