Zipf's Law in Passwords

Ding Wang and Gaopeng Jian, Peking University
Xinyi Huang, Fujian Normal University
Ping Wang, Peking University

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 underlying distributions of passwords. By conducting linear regressions on a corpus of 97.2 million passwords (a mass of chaotic data), we, for the first time, show that Zipf's law perfectly (natively) exists in the most vulnerable part of user-generated passwords, and we further provide compelling evidence that this law is highly likely to hold in the remaining part of user-generated passwords. By figuring out the corresponding exact distribution functions, we investigate some *fundamental* implications of our observations for password policies and password-based cryptographic protocols (e.g., authentication, encryption and signature).

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 its correctness 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. In two of four cases, our metric outperforms Bonneau's α-guesswork in simplicity 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 security administrators to gain a precise grasp of the strength of their password datasets and to adjust the password policies more reasonably.

Categories and Subject Descriptors: C.4.6 [**Operating Systems**]: Security and Protection–Authentication

General Terms: Theory, Security, Metric

Additional Key Words and Phrases: Internet security, Passwords, Password-based protocol; Zipf's law

ACM Reference Format:
Wang, D., Jian, G., Huang, X., and Wang, P. 2015. Zipfs Law in Passwords. *ACM Trans. Info. Syst. Sec.* 1, 1, Article 1 (January 2015), 34 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

User authentication is the most essential security mechanism for networked 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 user authentication, protecting hundreds of millions of accounts on Internet-scale service providers. Recently, there have been countless attempts in proposing alternative schemes (e.g., graphical passwords [Zhu et al. 2014], single sign-on [Sun et al. 2013] and multi-factor authentication [Huang et al. 2014]) to dislodge passwords, yet passwords are more widely used and firmly entrenched than ever. Since passwords offer many advantages not always matched by other alternative authentication schemes [Bonneau et al. 2012; Zhao et al. 2015; Wang et al. 2015a] and moreover, the transition costs of replacing them can

Author's address: D. Wang and P. Wang, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China; G. Jian, School of Mathematical Sciences, Peking University, Beijing 100871, China; X. Huang, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China. Email: {wangdingg,demscimath}@pku.edu.cn; xyhuang81@gmail.com; pwang@pku.edu.cn.

DOI:http://dx.doi.org/10.1145/0000000.0000000

not be effectively quantified [Herley and Van Oorschot 2012], they are likely to persist and dominate the authentication systems in the foreseeable future.

Despite its ubiquity, password-based authentication is accompanied by the dilemma of generating passwords which are challenging for powerful attackers to crack but easy for common users to remember. Truly random passwords are difficult for users to memorize, while user-chosen passwords may be highly predictable [Yan et al. 2004]. In practice, users tend to gravitate towards weak passwords that are related to their daily lives (e.g., birthdays, phone numbers, lovers, friends and pet names) [Brown et al. 2004; Florencio and Herley 2007], which means these passwords are drawn from a rather small dictionary and thus are prone to offline/online guessing attacks.

To mitigate this notorious "security-usability" dilemma, various password creation policies have been proposed, e.g., random generation [Yan et al. 2004], rule-based [Bishop and V Klein 1995; Schechter et al. 2010], entropy-based [Burr et al. 2006; Burr et al. 2013] and cracking-based [Houshmand and Aggarwal 2012; Castelluccia et al. 2012]. They force newly created passwords to adhere to some 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. For example, the password `password$1` is deemed "Very Weak" by Dropbox, "Fair" by Google and "Very Strong" by Yahoo!

The above contradictory outcomes of password strength (for more concrete examples, see [de Carnavalet and Mannan 2014; Wang and Wang 2015]) are a direct result of the inconsistent password strength meters employed among different websites, which may in part 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 [Ur et al. 2012]. The work by [Adams and Sasse 1999; Inglesant and Sasse 2010] 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 un-friendly policies.

As a result, different types of application systems typically have quite different favors. For e-commerce sites like eBay, portals like Yahoo! and order accepting sites like Kaspersky, usability is very crucial because every login event is a revenue opportunity. Anything that undermines user experience impairs the success of the business. So they tend to have less restrictive password policies [Florêncio and Herley 2010]. On the other hand, it is of great importance to prevent attackers from illicitly accessing valuable resources on security-critical sites, e.g., 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`).

As different systems favor varied password policies, a number of critical issues arise: how can the password policy designers evaluate their policies? How can the administrators select the right policy for their systems? In addition, usually the users of a system (as well as its services) may dynamically change as time goes on, which highly leads to large variations in the password dataset after some period of time (e.g., one year) even though the password policy[1] stays the same, especially true for Internet-scale service providers. In this situation, the security administrators shall quantify the strength of passwords and may need to adjust the password policy. Either failing to notice the changes in the password dataset or conducting improper countermeasures may give rise to great (but subtle) security and usability problems as shown above. So a proper assessment of the strength of password dataset is essential, without which the security administrator is unable to determine the following important question: How shall the password policy be

[1] In this work, the terms "password policy" and "password creation policy" will be used interchangeably, while policies regarding lockout and expiration [Chiasson and van Oorschot 2015] are out of the scope of this paper.

adjusted? Or equally, shall the password policy be enhanced to improve security, kept unchanged or even relaxed a bit to get usability in return? In a nutshell, the core crux of designing and selecting an appropriate password policy or properly adjusting it lies in *how to accurately assess the strength of password datasets created under it*. Note that, in this work we presume that each existing authentication system has already adopted some password policy (e.g., [Houshmand and Aggarwal 2012; Castelluccia et al. 2012]), and its adjustment mainly involves changing some rules and the password strength threshold.

1.1. Motivations

Surprisingly, as far as we know, existing literature has not provided 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 password dataset are theoretically measuring its information entropy (e.g., [Burr et al. 2006]) and empirically estimating its "guessability" (e.g., [Kelley et al. 2012; Mazurek et al. 2013]). The former, however, is not based on empirical data and has been shown inaccurate [Weir et al. 2010], while the latter, which largely depends on the choices of the cracking algorithms, parameters, and input dictionaries [Ma et al. 2014; Dell'Amico et al. 2010], has too many uncertainties to accurately characterize the strength of a given dataset. Later on, Bonneau [2012b] introduced an ingenious statistical-based metric $G_\alpha(\cdot)$ (named α-guesswork) which is parameterized on an attacker's desired success rate α. This metric is accurate, yet it is intrinsically non-deterministic. For instance, the relationship of $G_{0.50}(A) > G_{0.50}(B)$ can never ensure that $G_{0.49}(A) > G_{0.49}(B)$, where A and B are two password datasets. This means no definite conclusions can be reached unless an entire α-guesswork curve (with x-axis ranging from [0,α]) is computed. Failing to catch this subtlety may cause great misconceptions as it did in the case of [Li et al. 2014]. This non-deterministic nature undermines the simplicity of α-guesswork. Fortunately, in this work we develop a simple, accurate and deterministic (in two of four cases) statistical metric.

Inevitably, the accomplishment of accurately assessing the strength of a password dataset would entail the settlement of a more fundamental question: How to precisely characterize a given password dataset? Or equally, *what's the distribution that real-life user-generated passwords follow*? Despite more than thirty years of intensive research efforts, textual passwords are still enveloped in mysterious veils and this same old question is asked year in year out, which may well explain why most of today's password authenticated key exchange (PAKE) protocols with provable security (in thousands, some notable ones include [Abdalla et al. 2015a; Chen et al. 2014] in the random oracle model and [Yi et al. 2014; Katz and Vaikuntanathan 2013] in the standard model) still rely on a simple but inconceivable assumption: Passwords follow a uniform distribution.

To the best of knowledge, the work by Malone and Maher [2012] may be the most relevant to what we will discuss in this paper. They made an initial attempt to investigate the distribution of passwords and reached the conclusion that their password datasets are "unlikely to actually be Zipf distributed". Such a conclusion is right contrary to what we will show in the current work. Malone and Maher [2012] also concluded that "Zipf distribution is a relatively good match for the frequencies with which users choose passwords". A bit self-contradictory? The key is that, they use an inherently flawed method to attempt to model password distribution with Zipf (naturally, they fail), and they compare their model with a uniform model, and their comparison results show that their model is "a relatively good match". Since nearly any model would outperform a uniform model, the conclusion that their model is "relatively good" is of little, actually no, sense. This confusing, unsatisfactory situation motivates this work.

1.2. Our contributions

In this work, we bring the understanding of real-life passwords and the evaluation of password datasets onto a sound scientific footing by adapting statistical techniques, and seek to provide compelling answers regarding the above-mentioned two fundamental questions: (1) *What's the underlying 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 show that Zipf's law exists in real-life passwords, inspired by the applicability of Zipf's law to describe surprisingly diverse natural and social phenomena, such as the Internet topology [Faloutsos et al. 1999] and US firm sizes [Axtell 2001]. We prune the least frequent passwords, rank the frequency of each remaining unique password in decreasing order and investigate the mathematical relationships between the frequency and the rank by using linear regression. Extensive experiments on a massive corpus of twelve password datasets (vastly different in terms of service, size, how leaked, localization and language) show that, our Zipf model is able to accurately characterize the distribution of real-life passwords. More specifically, we show that: (1) the vulnerable portion of user-chosen passwords (i.e., popular passwords such as ones with a frequency $f_r \geq 5$) natively follow a Zipf-distribution; and (2) that the remaining portion of user-chosen passwords (i.e., un-popular passwords such as ones with a frequency f_r <5) is highly likely to follow a Zipf-distribution. This invalidates the claim made in [Malone and Maher 2012] that user passwords are "unlikely to actually be Zipf distributed". Particularly, we show it is the front head of passwords *natively* follow the Zipf's law, but not "the long tail of password choices" as reported in their work. Then, we figure out why such diametrically opposite observations are made and why our methodology is essential. Furthermore, we demonstrate the general applicability of our observation, highlight its fundamental implications for password policies and password-based cryptographic protocols and suggest a reasonable answer to the open problem left by Wang et al. [2015a]. This constitutes a compelling answer to the first question.

Our second contribution is a novel metric that utilizes the *concrete* knowledge of the password distribution function, and thus it overcomes various problems in existing metrics (e.g., uncertainties in cracking-based approaches [Kelley et al. 2012] and non-deterministic nature in α-guesswork [Bonneau 2012b]). Our metric facilitates the password policy designers and security 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. This suggests the settlement of the second question.

Another contribution of this paper is to show the effectiveness of our metric for measuring the strength of password datasets through empirical evidence. Firstly, we simulate optimal guessing attacks on the collected real-life password datasets. Then, we take a step forward to employ the state-of-the-art cracking algorithm (i.e., Markov-based [Ma et al. 2014]) to approximate optimal password cracking attack. Of independent interest may be our observation that, in some cases, Markov-based cracking success rates are much lower as compared to optimal cracking results, implying that the state-of-the-art cracking algorithm is far from an optimal one and there leaves much room for future improvement. Additionally, we report an inherent flaw in the strength conversion of α-guesswork [Bonneau 2012b] and manage to figure out how to fix it.

Roadmap. In Section 2, we survey related works. Then, we show Zipf's law exists in passwords in Section 3. Some fundamental implications of our observations are discussed in Section 4. The password dataset strength metric is presented, proved, and empirically established in Section 5, and 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 background 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" [Klein 1990]. The criteria can be divided into two types. One type is the exact rules for what constitutes an acceptable password, such as minimum length 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 is indeed the same as password creation policies we know today, and thus in this work 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 [Spafford 1992b] and ProCheck [Bergadano et al. 1998]. 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 [Burr et al. 2006]. However, by modeling the success rates of current password cracking techniques against real-life user passwords created under different rules, Weir et al. [2010] showed that merely rule-based policies perform poorly for ensuring a desirable level of security. On the basis of Weir et al.'s work, Houshmand and Aggarwal [2012] proposed a novel policy that first analyzes whether a user selected password is weak or strong according to the empirical cracking-based results, and then modifies the password slightly if it is weak to create a strengthened password. This policy facilitates measuring the strength of individual passwords more accurately and in addition, it can be adjusted more flexibly than previous policies due to the fact that its adjustment only involves tuning the threshold within a continuous range.

Perhaps the most relevant policy related to our strength metric for assessing password datasets (see Section 5) is suggested by Schechter et al. [2010]. 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 with 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 $\mathcal{T}$ (e.g., $\mathcal{T} = \frac{1}{10^6}$ as exampled in [Schechter et al. 2010]), yet whether it would greatly reduce usability has not been evaluated thoroughly (e.g., no actual user case studies are reported). As an immediate consequence of this policy, it might frequently annoy users by forbidding them to use their intended passwords that are typically popular. For instance, about 34.89% of users in www.tianya.cn use passwords that are more frequent than $\mathcal{T} = \frac{1}{10^6}$, which indicates that more than one third of the users have an equal potential to be annoyed to select and maintain a new password. Nevertheless, 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, such as on-line guessing, offline guessing, keylogging, shoulder surfing and social engineering [Long 2011; Herley 2013]. Here we only consider the on-line and offline guessing attacks, other attacks are unrelated to password strength or password dataset strength and therefore outside the scope of this work. While online guessing can be well thwarted by non-cryptographic techniques, such as locking an account after a threshold number of failed logins or using more flexible lockout strategies [Van Oorschot and Stubblebine 2006; Alsaleh et al. 2012], offline guessing attacks are performed on local hardware that the attacker controls and thus she can make as many guesses as possible given enough time and computational power. Florêncio et al. [2014] discussed scenarios where offline guessing constitutes a real threat and identified a great "chasm" between a password's guessing-resistance against these two types of guessing. They found that in this "chasm", incrementally increasing the strength of passwords delivers little security benefit, and thus they called in question the common practice of nudging users towards stronger passwords beyond online guessing resistance. Yet, it is not difficult to see that such a "chasm" would be largely eliminated (and so is the corresponding doubt), if one considers the cases where passwords (e.g., in salted-hash) have been leaked yet this leakage is detected (and coped with) only after some period of time (e.g., a few days), during which offline guessing indeed poses a realistic threat.

Consequently, it is essential for password-based authentication 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., the number of guesses) against the percentage of hashed passwords that would be offline recovered.* This measure only depends on the attacking technique and the way users choose their passwords, and it is neither related to the particular nature of the authentication system (e.g., which type of hash function is used, PBKDF2 or SHA-1?) 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 [Dell'Amico et al. 2010]. For example, system countermeasures against offline attacks, such as salting to defeat pre-computation techniques (e.g., Rainbow tables [Oechslin 2003]) or key strengthening [Dürmuth 2013] 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 concrete 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 passwords (e.g., words from dictionaries like the famous "dict-0294" [Outpost9.com's Lab 2014] or something related to their daily lives) that are easily rememberable [Shay et al. 2010; Florencio and Herley 2007]. However, users rarely use unmodified elements from 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 `pa$$word` is generated by leeting two letters of 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 "dict-0294" [Outpost9.com's Lab 2014], and this sort of techniques has emerged as early as 1979 in Morris-Thompson's analysis of 3,000 passwords [Morris and Thompson 1979]. This initial work has been followed by independent works by Klein [1990] and Spafford [1992a]. Later on, some dedicated software tools like John the Ripper [Designer 1996] appeared. Subsequent studies (e.g., [Kuo et al. 2006; Dell'Amico et al. 2010]) 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. Narayanan and Shmatikov [2005] developed an advanced cracking algorithm that uses Markov chain instead of ad hoc mangling rules to model user password creation patterns. This algorithm generates passwords that are phonetically similar to words. It 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 (PCFG), Weir et al. [2009] suggested 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, `pa$$word123` is denoted by $\mathsf{L}_2\mathsf{S}_2\mathsf{L}_4\mathsf{D}_3$. Then, a set of word-mangling rules is obtained from a training set of clear-text passwords. To simulate the optimal attack, this algorithm generates guesses in decreasing order of probability, and it is able to crack 28% to 129% more passwords than John the Ripper [Designer 1996]. In 2014, Ma et al. [2014] found that, when tuned with the right order and employing some ways to deal with the problems of data sparsity and normalization, Markov-chain based cracking algorithms would perform better than PCFG-based cracking algorithms. Hence, in this work we follow Ma et al.'s Markov-based algorithms 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, and then describe the collected datasets. Further, we provide a foundational understanding of passwords and show that Zipf's law perfectly exists in real-life passwords.

3.1. Linear regression

In statistics, linear regression is an approach for modeling the relationship between two variables by fitting a linear equation to the observed data. One variable is considered to be an explanatory variable, and the other one is considered to be a dependent variable. Usually, linear regression refers to a model in which, 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. 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 the 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, and similarly 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*. A R^2 value of 1 indicates that all data points perfectly dwell on the regression line.

3.2. Description of the password datasets

In our work we collect twelve large-scale real-life password lists different in terms of service, size, how leaked, user localization, language and culture (faith) background, showing that our model is able to accurately characterize the distribution of real-life passwords. All twelve datasets, as summarized in Table I, were compromised by hackers or leaked by anonymous insiders, and were subsequently disclosed publicly on the Internet. Some of them have also been used by a number of scientific works that study passwords (e.g., [Weir et al. 2010; Komanduri et al. 2014; Ma et al. 2014]). 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 I. Basic information about the twelve datasets ("PWs" stands for passwords)

Dataset	Service	Location	Language	When leaked	How leaked	Total PWs	Unique PWs
Tianya	Social forum	China	Chinese	Dec. 4, 2011	Hacker breached	30,233,633	12,614,676
Dodonew	Gaming& Ecommerce	China	Chinese	Dec. 3, 2011	Hacker breached	16,231,271	11,236,220
CSDN	Programming	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	Phishing attack	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
Rockyou	Gaming	USA	English	Dec. 07, 2009	SQL injection	32,603,388	14,341,564
Yahoo	Web portal	USA	English	July 12, 2012	Hacker breached	453,492	342,515
Mail.ru	Email	Russia	Russian	Sep. 09, 2014	Phishing&malware	4,938,663	2,954,907
Yandex.ru	Search engine	Russia	Russian	Sep. 09, 2014	Phishing&malware	1,261,810	717,203

The first four datasets, namely Tianya, Dodonew, CSDN and Duowan, are all from Chinese websites. We name each password dataset according to the corresponding website's 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 [Martin 2012] 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 4.98 million plain-text ones.

The fifth 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 social engineering (i.e., 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 [Bowes 2011] which contained 41,545 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 is 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 9,709 passwords.

The eighth dataset is from www.hak5.org and it was compromised by a group called ZF0 (Zero for 0wned) [Constantin 2009]. 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, homebrew, forensics, and network security", its dataset is amongst the weakest ones (see Section 5.1) of all the datasets. In this work, we use this dataset as a *counterexample* for representatives of real-life password distributions.

Besides the above eight datasets, we additionally employ four datasets (i.e., Rockyou, Yahoo, Yandex.ru and Mail.ru) to show the generalizability of our findings of Zipf's law in Section 3.4 and 3.5, and due to space constraints, they will not be analyzed elsewhere. The Rockyou dataset includes 32M passwords leaked from the gaming forum Rockyou in Dec. 2009 [Allan 2009]; The 450K Yahoo passwords was made online by the hacker group named D33Ds in July 2012; The last two deatasets (i.e., 4.9M Mail.ru and 1.3M Yandex.ru) were leaked by Russian hackers in Sep. 2014, and about 90% of them are active [Mick 2014], and it is said that these credentials are collected not by hacking the sites but through phishing and other forms of hacking attacks on users (e.g., key-loggers).

3.3. Statistics about the password datasets

In this subsection, we report some statistical information about the collected datasets. Firstly, the character composition information is summarized in Table II. It is interesting to note that Chinese users are more likely to use only digits to construct their passwords, while English users prefer using 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 tend to generate their passwords by adding the digit "1" to a sequence of lower-case letters.

Table II. Character composition information about each password dataset

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

Table III shows the length distributions of each dataset. We can see that the most popular password lengths are between 6 and 10, which on average accounts 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 passwords of length 6 and 7 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 characters, 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 [Johnston 2013], and a plausible reason may be that the shift to longer allowed password lengths is a non-trivial issue.

Table III. 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 IV, 123456 takes the lead. It is a long-standing problem that a significant fraction of users prefer the same passwords as if by prior agreement, which 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., [Castelluccia et al. 2012]) will ultimately eliminate this issue. Most of the top 10 Chinese passwords are sole digits, while most of the top 10 English passwords are sole letters.

Table IV. 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
Top 3 (%)	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
Top 10 (%)	7.42%	3.28%	10.44%	6.78%	0.78%	3.40%	2.17%	7.20%

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. Faith also has a role in shaping user passwords. For example, the password jesus1 emerges in the top-10 lists of both Sigle.org and Faithwriters (which are sites for Christians). Startlingly, for several datasets a mere of top 3 most popular passwords account for more than 5% of all the passwords. This indicates that, to break into these corresponding sites, 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

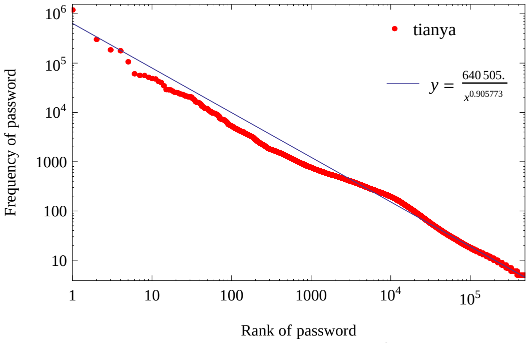


Fig. 1. Zipf's law in Tianya ($R^2 = 0.994$)

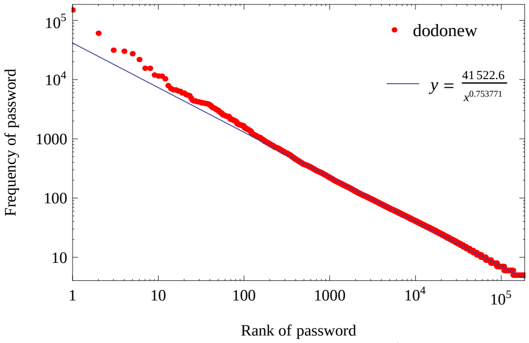


Fig. 2. Zipf's law in Dodonew ($R^2 = 0.996$)

dataset is still very concentrated and as we will show later in Section 5.1, it is among the weakest ones. This means that seemingly complex passwords may not be difficult to crack and actually may be rather weak, which further suggests the necessity and importance of a foundational understanding of passwords.

3.4. Zipf's law in passwords

Initially, PCFG is a machine learning technique used in natural language processing (NLP), yet Weir et al. [2009] managed to exploit it to automatically build password 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. [2013] and in dealing with the sparsity problem in passwords by Ma et al. [2014].

Inspired by these earlier works, in this study we make an attempt to investigate whether the Zipf's law,[2] 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 [1949]. It states that given some corpus of natural language utterances, the frequency of any word in it is inversely proportional to its rank in the frequency table. More specifically, for a natural language corpus listed in decreasing order of frequency, the rank r of a word and its frequency f_r are inversely proportional, i.e. $f_r = \frac{C}{r}$, where C is a constant depending on the particular corpus. This means that the most frequent word will occur about two times 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 Internet topology [Faloutsos et al. 1999], US firm sizes [Axtell 2001] and distribution of Linux software packages [Maillart et al. 2008].

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 $\mathcal{DS}$, the rank r of a password and its frequency f_r follow the equation

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

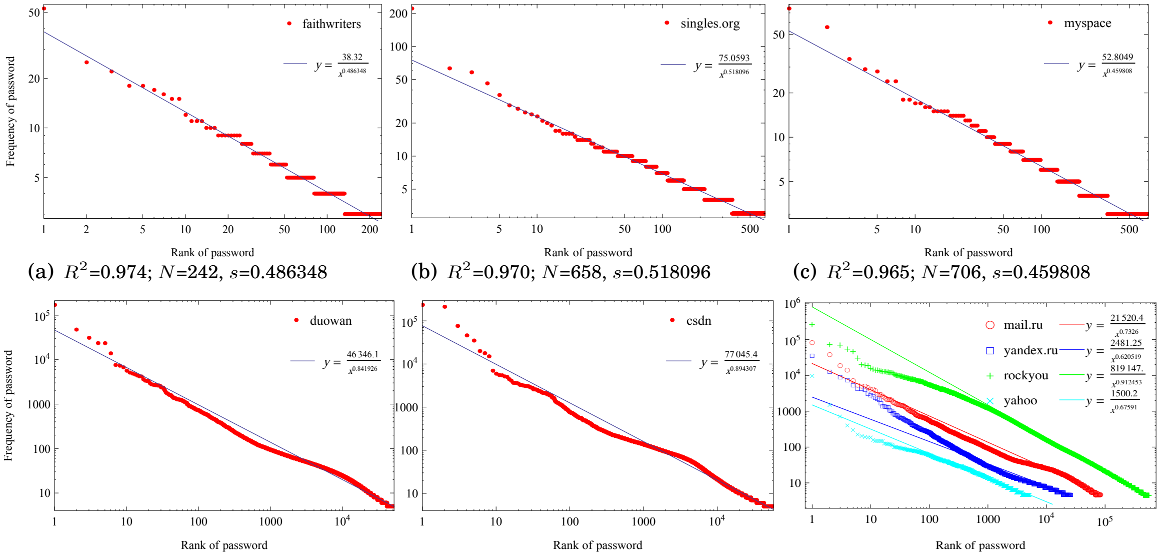
where C and s are constants depending on the chosen dataset, which in turn is essentially determined by many confounding factors (such as the type of web services to be protected, the underlying password policy adopted by the site, and the demographic factors of users like age, gender, educational level and language). Zipf's law can be more easily observed by plotting the data on a log-log graph (base 10 in this work), with the axes being log(rank order) and log(frequency). In other words, $\log(f_r)$ is linear with $\log(r)$:

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

[2] 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 [Adamic 2014].

As can be seen from Fig. 1, 30.23 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 Fig. 2 and the miniatures in Fig. 3, passwords from the other ten 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 V, every linear regression (except for Hak5) is with its R^2 larger than 0.965, which closely approaches to 1 and thus indicates a remarkably sound fitting. As for "Hak5", its R^2 is about 0.923, which is, though acceptable, 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 should also be noted that, how the datasets leak may have a direct effect on R^2. As can be confirmed by Table V, datasets leaked by phishing attacks are likely to have a lower R^2 as compared to those of datasets leaked by website breaches, because phishing attacks are unlikely to obtain the entire dataset of a website, while website breaches, once succeed, all (or at least an overwhelming part of) passwords of the website will be harvested.



(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) Zipf's law in four additional datasets

Fig. 3. Zipf's law in real-life passwords plotted on a log-log scale

The reason why we need to prune the least frequent passwords will be elaborated in Section 3.5. The selection of a specific small value (e.g., 3 or 5) as the threshold of least frequency (LF) is essentially based on the findings in statistics that (see Fig. 3 of [Clauset et al. 2009]): when *the sample size* is smaller than *the sample space*, the regression first improves greatly as LF progressively increases until reaching the best point $\hat{p}$, after which the regression deteriorates (because of dwindling the sample size) extremely slowly as LF increases. We have performed a series of experiments to identify the exact LF that enables the regression to reach $\hat{p}$, and find that, as a rule of thumb, for large datasets with millions of passwords, one can set $LF = 5$, otherwise set $LF = 3$. More complex ways (see pp.12 of [Clauset et al. 2009]) might be employed to estimate this threshold and to more accurately determine the distribution parameters, yet they are out of the focus of this work. Nonetheless, the regression results in Table V demonstrate that our selection of the least frequency threshold is satisfactory: every regression attains a R^2 close to 1.

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 that: (1) there is a close linking between N and the total passwords — the larger N is, the larger the latter will be; (2) the parameter s falls in the range [0, 1], which is different from other natural/social phenomena (e.g., intensity of solar flares, intensity of wars and frequency of family names [Newman 2005]) that are with $s > 1$.

We emphasize that, the exclusion of these least frequent passwords is inherently due to the limited sizes of the samples available: though the password population (i.e., the entire human-chosen passwords) perfectly follow a Zipf-distribution, the million-sized samples (e.g., 30M Tianya and 32M Rockyou) are still *too small to wholly exhibit this intrinsic feature*. This will be justified in Section 3.5. We also conjecture that it is only these popular passwords that will affect (reduce) the strength of a dataset, which will be established by both rigorous proofs and extensive empirical experiments in Section 5. In addition, to qualify as a proper description of a dataset, a distribution function $f(x)$ shall hold within a range $x_{min} \leq f(x) \leq x_{max}$ of at least $2 \sim 3$ orders of magnitude (i.e., $x_{max}/x_{min} \geq 10^{2 \sim 3}$) [Maillart et al. 2008]. Except for Hack5, this condition is satisfied by all our regressions.

Table V. Linear regression (LR) results of twelve password datasets ("PWs" stands for passwords)

Dataset	Totoal PWs	Least freq.	Fraction of PWs in LR	Unique PWs in LR (N)	Absolute value of the slope (s)	Zipf regression line (log$y$)	Coefficient of determination(R^2)
Tianya	30,233,633	5	0.50443286	486,118	0.905773	$5.806523 - 0.905773{*}\log x$	0.994204954
Dodonew	16,231,271	5	0.21640911	187,901	0.753771	$4.618284 - 0.753771{*}\log x$	0.995530686
CSDN	6,428,287	5	0.29841262	57,715	0.894307	$4.886747 - 0.894307{*}\log x$	0.985106832
Duowan	4,982,740	5	0.28653592	51,797	0.841926	$4.666012 - 0.841926{*}\log x$	0.976258449
Myspace	41,545	3	0.08094836	706	0.459808	$1.722674 - 0.459808{*}\log x$	0.965861431
Singles.org	16,250	3	0.22135384	658	0.518096	$1.875405 - 0.518096{*}\log x$	0.970277755
Faithwriters	9,709	3	0.12472963	242	0.486348	$1.583425 - 0.486348{*}\log x$	0.974175889
Hak5	2,987	3	0.15400067	76	0.643896	$1.579116 - 0.643896{*}\log x$	0.922662999
Rockyou	32,603,388	5	0.49600581	563,074	0.912453	$5.913362 - 0.912453{*}\log x$	0.997298647
Yahoo	453,492	3	0.22668537	12,608	0.675910	$3.176150 - 0.675910{*}\log x$	0.983232690
Mail.ru	4,938,663	5	0.33034872	83,914	0.732600	$4.332851 - 0.732599{*}\log x$	0.970047769
Yandex.ru	1,261,810	5	0.34210777	26,003	0.620519	$3.394671 - 0.620519{*}\log x$	0.972507203

3.5. Justification for our methodology

Malone and Maher [Malone and Maher 2012] have also attempted to investigate password distributions. Yet contrary to our findings that user-generated passwords are Zipf distributed and that it is the popular passwords (i.e., *the front head* of the whole passwords) that *natively* follow the Zipf's law, they concluded that their datasets (including 32M Rockyou) are "unlikely to actually be Zipf distributed" and 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 different observations — they fitted all the passwords of a dataset to the Zipf model.

Unpopular passwords (e.g., with $f < 3$) constitute a non-negligible fraction of each dataset (see Table V) and become the long tail of password choices (see Fig. 1 of [Malone and Maher 2012]) or the "noisy tail" in the statistical domain [Newman 2005], yet they fail to reflect their true popularity according to the law of large numbers. More specifically, for a given password pw_i, each observation can be seen as a random Bernoulli variable with mean $\mu = p_{pw_i}$ and standard deviation $\sigma = p_{pw_i}(1 - p_{pw_i})$ [Bonneau 2012b], where p_{pw_i} is the *true probability* of pw_i. After $|\mathcal{DS}|$ samples, pw_i's *empirical probability* $\frac{f_{pw_i}}{|\mathcal{DS}|}$ is a binomial-distributed random variable with μ=p_{pw_i} and σ=$\sqrt{\frac{p_{pw_i}(1-p_{pw_i})}{|\mathcal{DS}|}}$, where f_{pw_i} is the frequency of pw_i in the password dataset $\mathcal{DS}$. Because generally 1-p_{pw_i}≈1, this gives a relative standard error (RSE):

$$\frac{\sigma}{\mu}=\sqrt{\frac{p_{pw_i}(1-p_{pw_i})}{|\mathcal{DS}|}}\cdot\frac{1}{p_{pw_i}}\approx\sqrt{\frac{f_{pw_i}}{|\mathcal{DS}|^2}\cdot\frac{|\mathcal{DS}|}{f_{pw_i}}}=\sqrt{\frac{1}{f_{pw_i}}}$$

This means that the *true probability* p_{pw_i} can be well approximated by the *empirical probability* $\frac{f_{pw_i}}{|\mathcal{DS}|}$ only when f_{pw_i} is relatively large. For instance, we can ensure a RSE$<\frac{1}{2}$ when f_{pw_i}>4 and a RSE$>\frac{1}{\sqrt{3}}$ when f_{pw_i}<3. Thus, these unpopular passwords will greatly *negatively* affect the goodness of fitting when the entire dataset is used in regression. This well explicates why diametrically opposed observations are made between [Malone and Maher 2012] and this work, and this also provides a *direct* reason for the necessity of pruning the unpopular passwords.

We observe that there exists a more essential (yet subtle) reason: even if the password population perfectly follows a Zipf-distribution, the million-sized samples (e.g., 30M Tianya and 32M Rockyou) are still *too small to wholly exhibit this intrinsic feature*. For example, www.csdn.net adopts a policy that allows passwords consisting of letters and numbers and with a length 8 to 16, which means that a user's password (denoted by a stochastic variable X) will have about $|X|$=$62^{16}-62^{8}$≈4.8∗10^{29} possible (distinct) values under this policy. But we have only got 6.42∗10^6 CSDN passwords from the leakage, a very small sample relative to $|X|$. Owing to the *polynomially decreasing nature of probability* in a Zipf distribution (see Eq.1), low probability events (e.g., with $f_r<3$) will overwhelm high probability events in a small sample, and thus such a small sample without exclusion of unpopular events is highly unlikely to reflect the true underlying distribution. It follows that, when fitting all passwords of relatively small datasets, the regression will be negatively affected by these unpopular passwords and no marked rule can be observed even if the front head of passwords exhibits a good Zipf property.

We emphasize that, though these least frequent passwords do not *natively* show the Zipf behavior, this fact does not contradict our assertion that *the password population (of a site) perfectly follows a Zipf distribution*. Table V shows that, generally, the larger the dataset is (or equally, the larger the sample size is), the larger the fraction of popular passwords (i.e., passwords used in regression) will be. Based on this trend, one can expect that, had the dataset been sufficiently large, unpopular passwords would be few and whether excluding them or not would have little impact on the goodness of the fitting. That is, the entire dataset will exhibit a Zipf property. Fortunately, one of our follow-up work (see http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/pin_zipf.pdf) on the distribution of human-chosen PINs, a special kind of passwords, well confirm this inference. One can see that, most of the examined 4-digit PIN datasets can be wholly fitted into a Zipf model — even if PINs with $f_r<10$ are excluded, there are still over 94% of the datasets left in the regression and they perfectly follow the Zipf's law ($R^2>0.97$).

To further justify our assertion that user-chosen password samples (i.e., datasets) follow Zipf's law, we investigate the regression behaviors of samples that are randomly drawn from a perfect Zipf distribution, and see whether these two types of samples show the same regression behaviors. We explore three parameters, i.e., exact distribution (3 kinds), sample size (8 kinds) and the least frequency concerned (5 kinds), that might influence a regression and thus perform a series of 120(=3·5·8) regression experiments. More specifically, suppose that the stochastic variable X follows the Zipf's law and there are N=10^3 possible values $\{x_1,x_2,\cdots,x_{10^3}\}$ for X. Without loss of generality, the distribution law is defined to be $\{p(x_1)=\frac{C/1^s}{\sum_{i=1}^{N}\frac{C}{i^s}}=\frac{1/1^s}{\sum_{i=1}^{N}\frac{1}{i^s}},\ p(x_2)=\frac{1/2^s}{\sum_{i=1}^{N}\frac{1}{i^s}},\cdots,\ p(x_N)\ =\frac{1/N^s}{\sum_{i=1}^{N}\frac{1}{i^s}}\}$, where the sample space N and the slope s define the exact Zipf distribution function. To be robust, each experiment is run 10^3 times; For better comparison, each experiment is with only one parameter varying. Due to space constraints, Table VI only includes 40 experiments where Zipf N is fixed to 10^3 and Zipf s to 0.9, the sample size varies from 10^2 to $2\cdot10^4$ and LF increases progressively from 1 to 5. Readers are referred to all 120 experimental results in [Wang et al. 2015b]. Note that some integral statistics (e.g., the fitted N) in Table VI are with decimals, because they are averaged over 1000 repeated experiments.

Table VI. Effects of sample size and least frequency (LF) on linear regression when simulating a Zipf distribution

Zipf N	Zipf s	**Sample size**	**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
1000	0.9	20000	1	995.527	20000.000	100.00%	995.527	0.994645	0.943878
1000	0.9	20000	2	995.514	19979.918	99.90%	975.432	0.968837	0.969613
1000	0.9	20000	3	995.521	19886.123	99.43%	928.450	0.938901	0.986291
1000	0.9	20000	4	995.550	19665.232	98.33%	855.099	0.912336	0.994446
1000	0.9	20000	5	995.544	19298.183	**96.49%**	763.027	**0.894282**	**0.997415**

Note: For more detailed results, readers are referred to the supplemental material [Wang et al. 2015b] of this work.

Our results on 120 experiments show that, given a Zipf distribution (i.e., when the Zipf parameters N and s are fixed), no matter the sample size is smaller than, equal to or larger than N, larger LF will lead to a better regression (i.e., the fitted s is closer to the Zipf s, and R^2 is closer to 1) at the beginning, but will worsen the situation as LF further increases. More specifically, when the sample size is *smaller* than N, the fitted s first increases and then decreases as LF increases progressively; When the sample size is larger than N, on the contrary, the fitted s first decreases and then increases as LF increases progressively. Thus, we can identify the best fittings (in bold) and from them we can see that, the larger the sample size, the larger the fraction of popular events will be used in regression. This behavior well complies with our observation on real-life password datasets.

Particularly, when the sample size is sufficiently large (e.g., $10^4 \gg N = 10^3$), popular events (e.g., with $f_r \geq 3$) invariably account for over 95% of each sample and perfectly follow Zipf's law ($R^2 \geq 0.99$). This behavior well agrees with our regressions on PINs and with our inference on password datasets. In addition, when the sample size is much smaller than the sample space N, unpopular events constitutes the majority yet we have to exclude them to obtain a good fitting. This justifies our methodology of data processing (i.e., pruning unfrequent passwords) when performing regression analyses, because the sizes of real-life password datasets are generally much smaller than the password sample space (e.g., $6.42 * 10^6 \ll |X_{csdn}| \approx 4.8 \cdot 10^{29}$). In a nutshell, all the behaviors shown in our regressions on twelve password datasets well accord with the 120 simulated experiments.

3.6. General applicability of our observations

In the regressions in previous sections, we only considered datasets that are generated under loose (or no) password creation policies, which provides a clear insight into the natural behaviours of human-beings (i.e., password behaviors with little external restriction and interference). Table II∼IV show that quite short and letter-only passwords appear in every dataset (though the proportion of such passwords is marginal in some sites like Duowan, CSDN and Myspace), which suggests that there is no evident length or composition requirement for generating passwords in any site. 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 stricter (or looser) rule(s) is applied; Sometime later, the sites were hacked.

However, this is not true in some cases, especially for security-critical services which may implement strict policies at the very beginning. To further establish the applicability of our findings, two special kinds of datasets created under more constrained (yet quite realistic) password policies are considered: (1) Datasets with password lengths satisfying some minimum length (e.g., at least length-8); and (2) Datasets with each password being a mix of letters and numbers (e.g., at least one letter and one number).

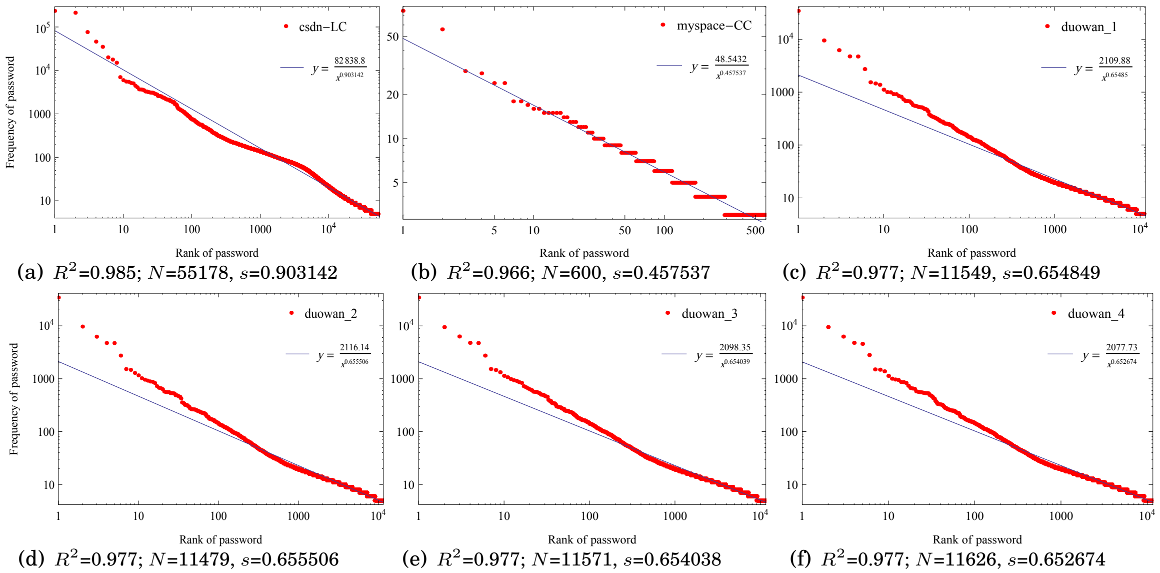
Since we did not have exact examples of passwords exactly generated under some specific creation policies with a length or composition requirement (as far as we know, there is no such ideal data publicly available), we attempted to model such policies by further dividing these datasets based on the minimum 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 caution 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" [Ur et al. 2012]. Mazurek et al. [2013] have also reported a similar observation. Fortunately, after careful examination of our twelve datasets (see Table II and Table III), we find that:

(1) Only 2.17% passwords in CSDN are shorter than eight characters long. These short passwords are highly due to the initial loose policy and the other remaining 97.83% long passwords are due to the later enhanced password policy. 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". This 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 constitute 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. 4(a) and 4(b), respectively. We can see that, the coefficients 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 twelve 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-value$\geq$0.05). Due to space constraints, only four randomly selected subsets (each with a size of 1 million) from Duowan are depicted in Fig. 4(c) $\sim$ Fig. 4(f). As R^2 of these four regressions are all 0.977 and very close to 1, it indicates Zipf's law fits well in these subsets. This implies that if we can obtain a sufficiently large subset of passwords of an authentication system, then the distribution of the whole passwords can be largely determined by conduction a linear regression and fitting them 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.



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

At this stage, a natural question arises: *Can our observations be generalized to user-generated passwords in most cases?* Or equally this question may be expressed as: Whether the datasets used in this work can be representative of most of the datasets? The answer is highly affirmative. On the one hand, the datasets used in this work are so far the most diversified (in terms of service, size, how leaked, locality, language and culture/faith) 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 knowledge, the most diversified datasets (i.e., three from US and three from China, and with each from different services) have been reported in [Ma et al. 2014] and the largest datasets (i.e., seven datasets with a total of 114 million passwords) have been used in [Li et al. 2014], while in our work we employ twelve diversified datasets with a total of 97.2 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 are leaked from sites with great importance (e.g., e-banking). Nevertheless, these datasets at least represent general credentials to a large extent, because it is well known that users tend to reuse passwords across multiple sites and even across sites of different categories. For instance, over one third of the 676 users polled by Sophos Ltd. admit that they use a single password across all their sites [Sophos Press Release 2009] and this figure is 43∼51% as reported in [Das et al. 2014]; Bailey et al. [2014] found that about 47% users reuse (including both exact and approximate reuse) their financial passwords on low value accounts (e.g., social sites). Consequently, our twelve datasets still represent a significant number of diversified user-generated passwords and can be employed to investigate general password distributions, and their identified striking property is unlikely to be a coincidence due to the high coefficients of determination.

On the other hand, rigorously speaking, there is no (or never will be) *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 whether 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 whether our theory is definitely correct; With adequate data and right tools, we can only develop models to characterize password distributions more and more accurately, and this might be a never-ending work in progress. We frankly admit more efforts need to be devoted to this interesting 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.

4. SOME FOUNDATIONAL IMPLICATIONS

In this section, we show two foundational implications of our Zipf theory. We believe this theory is also of interest in other domains, and it lays the foundation for their further theoretical development and practical application (see "GenoGuard" [Huang et al. 2015]).

4.1. Implications for password policies

Recently, many works on password policy (e.g., [Schechter et al. 2010; Castelluccia et al. 2012]) have suggested disallowing users from choosing dangerously-popular passwords (e.g., `123456` and `password123`) which occur with probabilities greater than a predefined threshold $\mathcal{T}$ (e.g., $\mathcal{T} = 1/10^6$). 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. So far, little underlying 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 choose popular passwords under a given threshold? What proportion of users will be affected if we restrict the top 0.0001% most popular passwords? How about restricting the top 0.01% most popular passwords?

We are now ready to answer these questions. In Section 3, we have shown that in most cases, user-generated passwords obey the 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 [Adamic 2014], 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 as well as the fact that $\alpha = 1 + 1/s > 2$ (see s in Table V), the largest frequency $x_{\mathcal{T}}$ allowed under a threshold $\mathcal{T}$ can be determined

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

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 $W_a(X > x_{\mathcal{T}})$,[3] 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

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

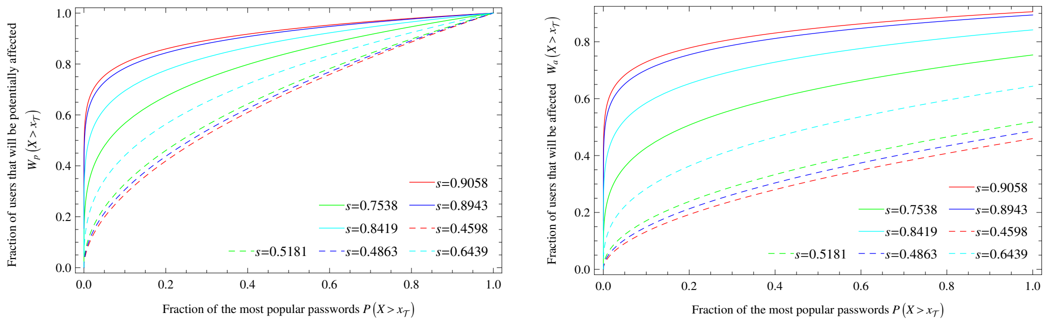
Similarly, Eq.9 can be re-written as

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

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 a better comprehension, in Fig. 5 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 V.

The steep increase of W_p and W_a at the very beginning of their curves (see Fig. 5) explicitly reveal that, popular passwords are overly popular and a non-negligible fraction

[3] 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 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) Users that will be *potentially* affected with $P(X > x_\mathcal{T})$ (b) Users that will be *actually* affected with $P(X > x_\mathcal{T})$

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

of users will be inconvenienced even if only a marginal 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 VII. 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" [Newman 2005] 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, and this will inevitably introduce some deviations.

Table VII. 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		$\mathcal{T}$ =1/1000000	
	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%	0.4416%	30.9110%
Dodonew	0.0001%	1.3926%	0.0009%	3.1556%	0.0014%	3.6298%	0.2958%	11.2351%
CSDN	0.0002%	9.4648%	0.0029%	12.2806%	0.0049%	12.8732%	0.8441%	24.6874%
Duowan	0.0004%	5.8130%	0.0048%	8.8648%	0.0079%	9.6064%	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: A dash "–" stands for "not applicable", due to the mere fact that $1/\mathcal{T}$ is larger than the size of corresponding dataset.

Though the above theoretical model is not perfectly accurate, as far as we know, it for the first time 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 security administrators to offer a more acceptable trade-off between usability and security. For example, under our theory it is not difficult to see that it might be unreasonable to set $\mathcal{T} = 1/10^6$ for Internet-scale sites, for more than 60% users will be potentially annoyed. However, Schechter et al. [2010] and Florêncio et al. [2014] just explicitly (or implicitly) suggested such a value for $\mathcal{T}$. On the other hand, the Zipf's law revealed in Section 3.4 suggests that the frequencies of the most popular passwords descend at an approximately logarithmic rate, and thus only a limited proportion of 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 of most systems will be annoyed when $\mathcal{T}$ is set to the moderate value $1/16384$ complying with a Level 2 certification [Burr et al. 2013], which suggests that $\mathcal{T} = 1/16384$

would be more acceptable for most Internet-scale e-commerce sites. 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. We also emphasize that the picture we draw here is an elementary, plausible (rather than conclusive) evaluation of the policy usability, and thorough field studies are still intrinsically necessary.

4.2. Implications for password-based authentication

Another foundational implication of our observation is for thousands of provably secure authentication protocols that involve passwords, like password-only single-factor schemes (e.g., two-party [Katz et al. 2009] and multi-party [Chen et al. 2014]) and password-based multi-factor schemes (e.g., two-factor [Wang et al. 2015a] and three-factor [Huang et al. 2014]). Here we first show the implication for password-only schemes, also called PAKE protocols. In most provably secure PAKE protocols (e.g., [Bellare et al. 2000; Canetti et al. 2012; Pointcheval 2012; Jarecki et al. 2014; Abdalla et al. 2015a; Chen et al. 2014] in the random oracle model and [Halevi and Krawczyk 1999; Katz et al. 2009; Katz and Vaikuntanathan 2013; Yi et al. 2014] 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 independent of the security parameter k", then a security model is described, and finally a "standard" definition of security as the one in [Katz et al. 2009] 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 $\mathcal{P}$."[4]

According to [Bonneau 2012b], user-generated passwords generally offer about $20 \sim 21$ bits of actual security against an optimal offline dictionary attack, which means the effective password space $\mathcal{D}$ is of size 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 and actually may be somewhat misleading in practice. For instance, the actual advantage of $\mathcal{A}$ against the gaming&e-commerce site www.dodonew.com reaches 1.49% when $Q(k){=}3$ and 3.28% when $Q(k){=}10$, which are far beyond the theoretic results given by Eq.13. Predictably, the advantages of $\mathcal{A}$ against most of the real-world sites will be largely underestimated, and an overly optimistic sense of security might be conveyed to common users and security administrators.

As a prudent side note, some of these works (e.g., [Katz and Vaikuntanathan 2013; Katz et al. 2009; Halevi and Krawczyk 1999]) 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, different distributions for different clients, or the password dictionary size depends on the security parameter. 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 the distribution would be if "user-chosen passwords do not distribute uniformly". This defeats the purpose of constructing provably secure protocols which "explicitly

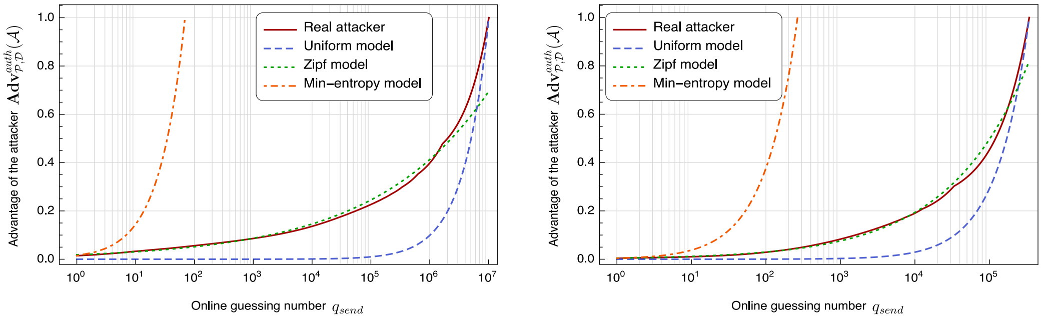
[4] We remark that some PAKE protocols (e.g., [Chen et al. 2014; Bellare et al. 2000]) relax this definition of security to $\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq c \cdot Q(k)/|\mathcal{D}| + \epsilon(k)$, where c is a constant positive integer, indicating $\mathcal{A}$ now is allowed to guess c passwords per on-line attempt. However, this does not necessarily mean that the corresponding protocol $\mathcal{P}$ is actually subject to the threat that $\mathcal{A}$ can guess more than one password per on-line attempt, for this relaxation may be due to some technical reasons in reductionist proofs but not an inherent defect of $\mathcal{P}$.

capture the inherently *quantitative* nature of security, via a concrete or exact treatment of security" and "offer *quantitative* security guarantee" [Bellare 1999] in the first place.

According to our Zipf theory, now it is fundamentally unnecessary (unrealistic) to make an assumption of uniform distribution of passwords. Instead, it is more desirable to make the Zipf assumption about password distributions. Since system assigned random passwords are hardly usable [Shay et al. 2012], 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 Section 3.4. Under the Zipf assumption, it is naturally to reach that

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) = \frac{C/1^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \frac{C/2^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} + \cdots + \frac{C/Q(k)^s}{\sum_{i=1}^{|\mathcal{D}|}\frac{C}{i^s}} = \frac{\sum_{j=1}^{Q(k)}\frac{1}{j^s}}{\sum_{i=1}^{|\mathcal{D}|}\frac{1}{i^s}}, \qquad (14)$$

Fig. 6 shows that $\mathcal{A}$'s advantage is more accurately captured by our Zipf model than the uniform-based model. The latter tends to greatly underestimates $\mathcal{A}$'s advantage of online guessing (i.e., when the guess number is small, e.g., less than 10^5). For instace, at 1000 guesses (i.e., $Q(k) = 10^3$), the uniform-based model estimates $\mathcal{A}$'s advantage against the Tianya service to be $7.75 * 10^{-8}$, yet the real value is 16.04%. Fortunately, our Zipf-based attacker estimates $\mathcal{A}$'s advantage to be 18.48%, well approximating the real value.



(a) Attacking the Dodonew service

(b) Attacking the Yahoo service

Fig. 6. With q_{send} (generally, $q_{send} \leq 10^5$) online guessing attempts, the advantages of the real attacker, our Zipf-modeled attacker, the uniform-modeled attacker and min-entropy-modeled attacker.

Due to the approaching-logarithm-function nature of harmonic numbers [Paule and Schneider 2003], the value on the right hand of Eq.14 will be alarmingly large. Thus, the system will be in serious danger even if $Q(k)$ 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 IV, seven of eight websites have a chance of more than 3.28% of accounts being breached by an attacker who makes merely ten online impersonation attempts.

From Eq.12 (or Fig. 6(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_p), which suggests that an online guessing attacker can succeed with a chance W_p just by trying $P{\cdot}|\mathcal{D}|$ different passwords. That is, 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 policies that restrict the overly popular passwords [Schechter et al. 2010]) shall be taken, which may lead to passwords with *a skewed Zipf distribution*. In this case, the skewed Zipf distribution seems hardly possible to be rigorously (mathematically) characterized, we are stuck in a conundrum to formulate the security result $\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k)$. Inspired by the essential notion of security that a secure PAKE protocol can provide – only online impersonation attacks are helpful to the adversary in breaking the security of the protocol [Halevi and Krawczyk 1999], we manage to get out of the problem by giving up the idea of firstly characterizing the distribution of password and then formulating the definition of security. And instead, whenever a popularity-based password policy like [Schechter et al. 2010] is in place, we provide a tight upper bound for the adversary's advantage. More specifically, Eq.13 now is amended as follows:

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq F_1 \cdot Q(k)/|\mathcal{DS}| + \epsilon(k), \tag{15}$$

where F_1, as said earlier, is the frequency of the most popular password in the dataset $\mathcal{DS}$, $|\mathcal{DS}|$ is the (expected) number of user accounts of the target authentication system, and the other notations are the same 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., $\mathcal{T}$ =1/16384) that the underlying password policy (see [Schechter et al. 2010]) maintains. For a system to reach a Level 1 certification [Burr et al. 2013], 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, for 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.15, where $F_1 = 1$ and $|\mathcal{DS}| = |\mathcal{D}|$.

We happen to find that a recent PAKE protocol proposed by Abdalla et al. [2015b] uses a different formulation of security from traditional ones:

$$\mathrm{Adv}_{\mathcal{A},\mathcal{P}}(k) \leq Q(k)/2^m + \epsilon(k), \tag{16}$$

where m is the *min-entropy* of the passwords.[5] Actually, it is not difficult to see that this kind of formulation (i.e., Eq.16) is in essential the same with Eq.15, for one can derive that $m = -\log_2(F_1/|\mathcal{DS}|)$ [Bonneau 2012b]. However, no rationale or justification for preferring Eq.16 but not Eq.13 is given in [Abdalla et al. 2015b]. In comparison, our formulation Eq.15 is more concrete and intelligible than Eq.16 from the prospective of password policy.

In addition, as with Eq.15, Abdalla et al.'s Eq.16 (i.e., the min-entropy model) is *only* effective when a popularity-based password policy like [Schechter et al. 2010] is in place, resulting in that the password distribution does not follow the Zipf's law. If no popularity-based password policy is in place, passwords will follow the Zipf's law, and Eq.16 (or equally Eq.15) will be meaningless. This has not been pointed out in [Abdalla et al. 2015b].

One can also see that, if m is defined to be the *entropy* of passwords, then Eq.16 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 Section 6.1 of [Benhamouda et al. 2013]) state that "equivalently the advantage of any adversary can be bounded" by either Eq.13 or Eq.16. However, as we have shown, if m is defined as the *min-entropy* of passwords (which is the right definition in most cases), Eq.13 and Eq.16 (or equally, Eq.15) will be significantly different from each other.

For better comprehension, we show in Fig. 6 how the uniform model, min-entropy model and our Zipf model approximate the real attacker when passwords follow a Zipf distribution (using Yahoo passwords and Dodonew passwords for example). Since online guessing attacks are generally prevented by lockout strategies [Florêncio et al. 2007], rate-limiting techniques [Alsaleh et al. 2012] or suspicious login detection [Dürmuth et al. 2016], the attackers cannot make a large number of login attempts, and thus the guess number is often small (e.g., $Q(k) \leq 10^3$). In this case, one can see that our Zipf model well

[5] We note that, in Sections 5.2∼5.4 of [Abdalla et al. 2015b], m is re-defined to be the *entropy* of passwords. This inconsistence would lead to great differences in security guarantees. We conjecture typos have occurred there.

approximates the real attacker who makes full use of the real password distribution, and it substantially outperforms the uniform model and min-entropy model. For instance, the actual advantage of $\mathcal{A}$ against Dodonew reaches 5.60% when $Q(k)$=10^2 and 8.59% when $Q(k)$=10^3, which are far beyond 0.00098% and 0.0098% given by Eq.13 and far less than 100% and 100% given by Eq.16, respectively. Fortunately, our Zipf model (i.e., Eq. 14) predicts a 5.41% when $Q(k)$=10^2 and a 8.63% when $Q(k)$=10^3. This is well accords the actual advantage of $\mathcal{A}$ against Dodonew.

Unlike PAKE protocols where users have to interact with the server to register their passwords, most multi-factor schemes provide a property, which is termed "DA2-Local-Secure" [Wang et al. 2015a], to facilitate users change their passwords freely and locally (i.e., without interacting with the server). Since there is no interaction with the 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 [Yang et al. 2008]), 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.[6] 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" [Wang et al. 2015a]), providing an answer to the open problem raised in [Wang et al. 2015a]: 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 our knowledge, we, for the first time, pay attention to the joint between passwords and password-based 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 have mainly taken password-based authentication as a case study, and one can easily find that our results revealed herein also can be readily applied to other kinds of password-based cryptographic protocols whose security formulation essentially relies on the *explicit* assumption of the distribution of passwords, such as password-based encryption (e.g., [Juels and Ristenpart 2014]), password-based signatures (e.g., [Gjosteen and Thuen 2012]) and password-protected secret sharing (e.g., [Jarecki et al. 2014]).

5. STRENGTH METRIC FOR PASSWORD DATASET

In this section, we address the question as to how to accurately measure the strength of a given password dataset. As one practical application of our Zipf theory, an elegant and accurate statistical-based metric on the strength of password datasets is suggested.

5.1. Our metric

Normally, a smart offline guessing attacker,[7] 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 [Dell'Amico et al. 2010; Bonneau 2012b].[8] Accordingly, we can use the cracking result $\lambda^*(n)$ to be the strength metric of a given password dataset:

[6] That said, when a popularity-based password policy is mandated, Eq.15 is more preferable.

[7] The attacks mentioned in this Section are all offline attacks, for our purpose is to measure the strength of an entire dataset, which is generally characterized by how much percentage of passwords in salted-hash (or unsalted-hash) could be successfully recovered (see Section 2.2).

[8] Note that, the optimal attack is of theoretic value (i.e., providing 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.

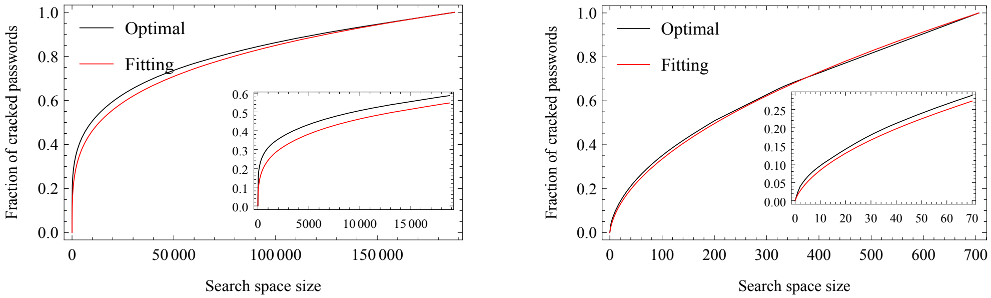
$$\lambda^*(n) = \frac{1}{|\mathcal{DS}|}\sum_{r=1}^{n} f_r, \tag{17}$$

where $|\mathcal{DS}|$ is the size of the password dataset and n is the number of guessing.

In Section 3, 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):

$$\lambda^*(n) \approx \lambda(n) = \frac{\sum_{r=1}^{n}\frac{C}{r^s}}{\sum_{r=1}^{N}\frac{C}{r^s}} = \frac{\sum_{r=1}^{n}\frac{1}{r^s}}{\sum_{r=1}^{N}\frac{1}{r^s}}. \tag{18}$$

It should be noted that, in Eq.18, $\lambda^*(n)$ is not exactly equal to the value of the 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.17 and $\lambda(n)$ as a function of n according to Eq.18, and put these two curves together to see how they agree with each other. In Fig. 7(a), we depict $\lambda^*(n)$ and $\lambda(n)$ for 30.23 million passwords from the Tianya dataset and obtain an average deviation of 1.32% (i.e., a sound fitting) for the two curves. Due to space constraints, here we cannot illustrate the related pictures for the other datasets like that of Tianya and Myspace, yet we summarize the average deviation between the two curves $\lambda^*(n)$ and $\lambda(n)$ ($1 \le n \le |\mathcal{DS}|$) for each dataset in Table VIII.



(a) Average deviation of Dodonew: 1.76%
(b) Average deviation of Myspace: 0.88%

Fig. 7. Consistence of optimal attack with our metric on two example datasets (Dodonew and Mysapce)

As evident from Table VIII, the $\lambda^*(n)$ curve well overlaps with the $\lambda(n)$ curve for each dataset. Specifically, except for Hak5, the average deviations are all below 2% (i.e., from 0.54% to 1.93%), suggesting sound consistence of $\lambda(n)$ with the optimal attacking result $\lambda^*(n)$. As with Fig. 7, the two curves for each dataset first deviate slightly when n is small and then gradually merge into each other as n increases. This is mainly caused by the deviation of the first *few* high-frequency passwords from the Zipf fitting line (see Fig. 3).

Table VIII. The average deviation between $\lambda^*(n)$ and $\lambda(n)$ ($1 \le n \le |\mathcal{DS}|$) for each dataset

	Tianya	Dodonew	CSDN	Duowan	Myspace	Singles.org	Faithwriters	Hak5
Average Deviation	1.32%	1.76%	1.93%	0.86%	0.88%	1.43%	0.54%	3.05%

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. Note that, essentially, measuring a password dataset is equivalent to measuring the policy under which this dataset is created. 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 security administrators now have a clearer picture about what level of strength their datasets can provide. The recent litany of catastrophic leakages of web accounts (see [Katalov 2013] for an incomplete list) provides wonderful materials to facilitate such comparisons.

THEOREM 5.1. *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 Section 5.2, and in Section 5.3 its compliance with cracking results will be shown by the simulated optimal attack and the state-of-the-art cracking algorithm (i.e., Markov-based [Ma et al. 2014]), respectively.

COROLLARY 5.2. *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 5.1.

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. This gives rise to four cases: (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 $\lambda_B(n)$ is parameterized on the variable n, and thus it is non-deterministic (i.e., unable to reach a direct conclusion). In such cases, we may have to draw the cure (search space n VS. success rate) with n ranging from 1 to N, similar to other methods such as the cracking-based approach (e.g., PCFG-based and markov-based [Ma et al. 2014]). Note that, in all four cases the statistical-based α-guesswork [Bonneau 2012b] is non-deterministic, i.e., it is *inherently* parameterized on the success rate α (e.g., a relationship of $G_{0.49}(A) > G_{0.49}(B)$ can never ensure that $G_{0.50}(A) \geq G_{0.50}(B)$). Bonneau [2012b] cautioned that "we can't rely on any single value of α, each value provides information about a fundamentally different attack scenario." In this light, our metric is more simple. We also identify and fix an inherent flaw in the strength conversion of α-guesswork, and the details can be found in Appendix A.1.

Some Remarks. Note that, as with the entropy metric recommended in the NIST SP800-63-2 document [Burr et al. 2013] and the α-guesswork proposed in [Bonneau 2012b], our metric is mainly effective on password datasets that are in clear-text or un-salted hash and cannot be applicable to passwords in salted-hash. This is an inherent limitation of all statistic-based metrics (e.g., [Burr et al. 2013; Bonneau 2012b] and ours). For salted-hash passwords, one needs to resort to attacking-based approaches (e.g., [Kelley et al. 2012; Ma et al. 2014]), albeit at the cost of reduced accuracy (as we will show in Section 5.3, attacking-based approaches in their current form have too many uncertainties and are far from ideal). 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 do not affect much the applicability of our metric 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, recent breaches clearly signal that Internet-scale sites with un-salted password storage are far from being rare exceptions. The most convincing evidence lies in the fact that most of the previously leaked datasets from prominent IT firms or leading organizations (such as Facebook, LinkedIn, Adobe, Dropbox, IEEE, to name just a few [Katalov 2013]) 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 the security provisions of its password policy. And our metric is the right choice. Thirdly, it is well known that the authorities in many countries (e.g., NSA of U.S.) have been asking Internet providers and websites to provide user password datasets (in plain-text) to them [McCullagh 2013]. In this case, these websites shall also maintain a copy of un-salted passwords to ensure compliance with the regulations. Last but not least, even if no plain-text (or unsalted-hash) passwords from real-life websites are available, field experiments (e.g., [Egelman et al. 2013]) can be used to collect user generated passwords. With these field passwords, our metric can be used to help password policy designers and security 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 practical in many realistic scenarios. In addition, as said earlier, in two of four cases, our metric has unparalleled advantage due to its deterministic feature than the state-of-the-art metrics (e.g., the attacking-based [Ma et al. 2014] and the statistic-based α-guesswork [Bonneau 2012b]). Yet, it is non-deterministic in the remaining two cases where we have to draw entire curves of $\lambda_A(n)$ and $\lambda_B(n)$, with n ranging from 1 to max$\{N_A, N_B\}$, which are quite similar to the “guessing curves” in the attacking-based approach [Ma et al. 2014] and α-guesswork [Bonneau 2012b]. We emphasize that, our metric is only workable and superior to these existing metrics when the underlying distribution obeys Zipf’s law, while in other cases (where password distribution deviates from Zipf) these existing metrics just come in handy.

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 $\sum_{r=1}^{N_A} \frac{C_A}{r^s} = 1, \sum_{r=1}^{N_B} \frac{C_B}{r^s} = 1$, and $C_A = \frac{1}{\sum_{r=1}^{N_A} \frac{1}{r^s}} < \frac{1}{\sum_{r=1}^{N_B} \frac{1}{r^s}} = C_B$. So when $1 \le n \le N_B$, we have

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

When $N_B + 1 \le n \le N_A - 1$, we can get

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

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

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

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

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

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

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

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

$$\begin{array}{ll} \lambda_A(n) < \lambda_C(n) & 1 \le n \le N_A - 1 \\ \lambda_C(n) < \lambda_B(n) & 1 \le n \le N_A - 1 \end{array}$$

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

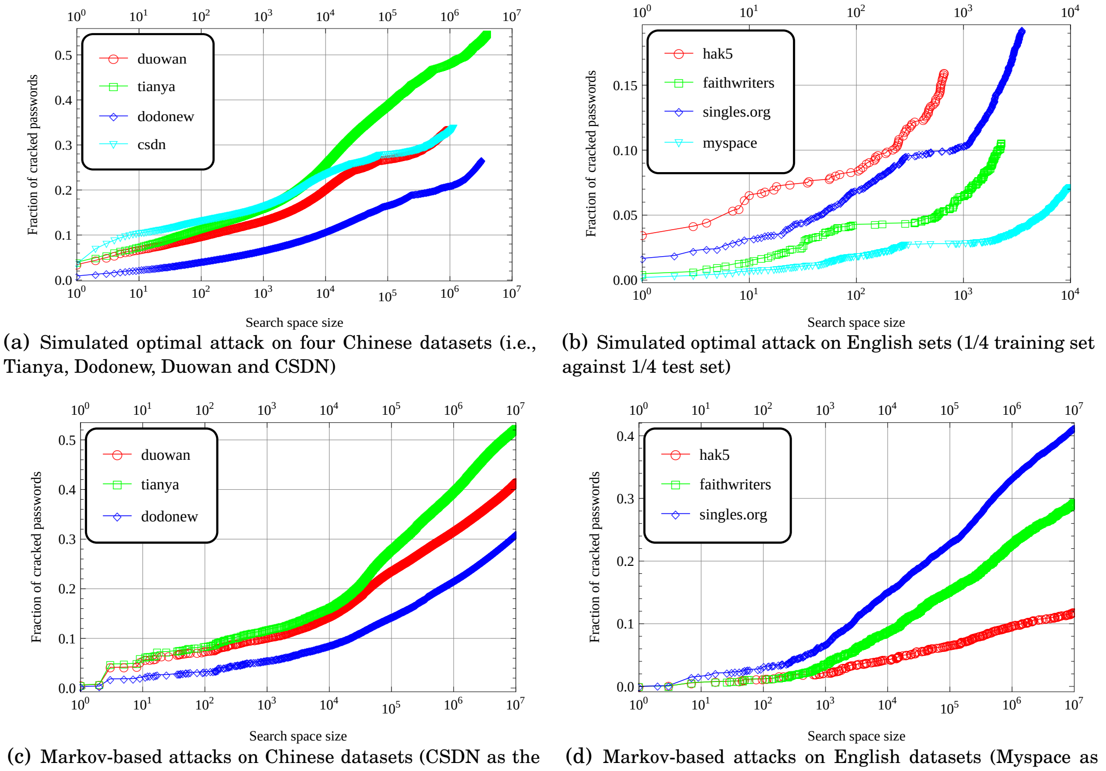
5.3. Experimental results

In this subsection, we further use the simulated optimal attack and the state-of-the-art password attacking algorithm on real-life passwords to show that our metric in Section 5.1 is practically effective. Since Markov-based cracking algorithms generally perform better than PCFG-based ones [Ma et al. 2014], here we prefer Markov-based algorithms.

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 authentication system. To see whether our metric is consistent with realistic attacks, we relax this assumption a bit and suppose that the attacker has obtained *a quarter* of the plain-text accounts (passwords) of the target system and use them to guess *one-third* of the remaining user accounts (which is another quarter of the total accounts) in any form (salted-hash or unsalted-hash). 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 accounts in plain-text. And thus this new attacking scenario is rather practical and we call it "simulated optimal attack". For better presentation, we divide the eight main datasets into two groups:[9] 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. 8(a), and results on group two are illustrated in Fig. 8(b). It is not difficult to see that, for any two datasets in the same group, the attacking results comply with our metric results listed in Table V. For instance, from Fig. 8(a) we know that, for any search space size (i.e., every n), dataset Duowan is weaker than dataset Dodonew, which implies $N_{\text{dodonew}} > N_{\text{duowan}}, s_{\text{dodonew}} < s_{\text{duowan}}$. This implication accords with the statistics in Table V.

Furthermore, we perform more realistic guessing attacks (i.e. Markov-based attacks) to assess the effectiveness of our metric. As in simulated optimal attacks, we divide the eight main datasets into two groups according to their sizes and languages. For the Chinese group, we use CSDN as the Markov training set; For the English group, we use

[9] As said earlier, due to space constraints the four auxiliary datasets (see Table I) are only shown to be Zipf-distributed, and actually, all the other general properties revealed in this work are also hold by them.



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

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

(c) Markov-based attacks on Chinese datasets (CSDN as the training set, backoff smoothing and end-symbol normalization)

(d) Markov-based attacks on English datasets (Myspace as training set, backoff smoothing and end-symbol normalization)

Fig. 8. Simulated optimal attacks and Markov-based attacks on different groups of datasets

Myspace as the Markov training set. As shown in [Ma et al. 2014], there are mainly three smoothing techniques (i.e., Laplace, Good-Turing and backoff) to address the data sparsity problem and two normalization techniques (i.e., distribution-based and end-symbol-based) to address the unbalanced password-length distribution problem. Ma et al. found that the attacking scenario that combines the backoff smoothing with the end-symbol based normalization performs the best, and thus we adopt this scenario. The cracking results for these two groups passwords are depicted in Fig. 8(c) and Fig. 8(d), respectively.

The test shows that the Markov-based attacking results on most of the datasets are consistent with our metric, and the only exception that violates our metric is on dataset Hak5. According to Table V, N_{Hak5} is smaller than that of any other datasets and s_{Hak5} is larger than that of any other datasets in the same group, which means Hak5 is the weakest one. However, Fig. 8(b) shows that, under the Markov-based guessing attack, Hak5 is the strongest among the three English test sets. This inconsistence may be because of its non-representative nature of a real password dataset, or due to the inappropriateness of our selected training set for the Markov-based guessing attack.

Of particular interest may be our observation that Markov-based attacks seem to be much less effective than simulated optimal attacks. For example, at 10^5 guesses, Markov-based attacks on Chinese datasets achieve success rates 14.5%$\sim$28.1%, lower 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 is due to the inherent weaknesses of cracking algorithms – their performance relies largely on the choices of training set, smoothing/normalization techniques and maybe external 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, use different training sets and specially choose smoothing/normalization techniques in our Markov-based experiments. This also highlights the intrinsic limitations of using empirical attacking results (e.g., [Weir et al. 2010; Kelley et al. 2012; Castelluccia et al. 2012]) as a strength measurement of password dataset, suggesting the necessity of our metric. In a nutshell, there is still room for developing more practical attacking algorithms that have fewer uncertainties yet are more effective.

6. CONCLUSION

In this work, we have provided a novel prospective of the distributions of 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 PAKE protocols (in thousands, some notable ones include [Chen et al. 2014; Canetti et al. 2012; Katz et al. 2009; Bellare et al. 2000]) have been proven secure under the hypothesis that passwords are uniformly distributed, yet we have shown that their formulations of security results fail to capture the actual advantages of real-life attackers and may have some unintended consequences. Accordingly, we suggest an amendment to more accurately characterize the formal security results of PAKE protocols.

Apart from its theoretical interest, we show a practical application of our Zipf theory by proposing a new statistic-based metric on the strength of password datasets. Our metric outperforms most of the existing statistic-based metrics in accuracy (e.g., [Burr et al. 2013]) and in two of four cases, even in simplicity (e.g., [Bonneau 2012b]). Of great interest is its *deterministic* nature of measurement of the dataset strength, which facilitates more simple and precise strength comparisons among different datasets. We have formally proved our metric in a mathematically rigorous manner and also fixed an inherent flaw in the strength conversion of α-guesswork [Bonneau 2012b]. We have taken a step further to evaluate the effectiveness of our metric by performing extensive cracking experiments on our large-scale corpus and demonstrated its practicality. We believe that the unveiling of this law is also of interest in other domains and that this work lays the foundation for their further theoretical development and practical application (e.g., the recent "GenoGuard" [Huang et al. 2015] relies on both our theoretical law and numerical results).

More work remains to be done on this interesting, important yet challenging topic. For example, what is the underlying mechanism that leads to the emergence of Zipf's law in a chaotic process like the user generation of authentication credentials? How will the password distribution of a system change (evolve) as time goes on? Do extremely high value accounts (e.g., e-banking passwords) obey Zipf's law? It is a mixed blessing that, the chances for such investigations to be conducted in the future are only increasing as more sites of high values are breached and more datasets are made publicly available.

There are also many other issues raised by the findings of this work, and they may entail comprehensive field studies. For instance, as we provide a sound rationale for the necessity of employing some popularity-based password policy, how should we set the popularity threshold? And, for a specific threshold, to what extent usability will be affected in practice? Is it necessary for multi-factor authentication protocols to give up the feature of supporting users in changing their passwords without interacting with the server? By applying the machinery of machine learning (e.g., linear regression), the Zipf theory promises to impart mathematical rigor to password use in system security (see Sec. 4.1). Meanwhile, the Zipf theory introduces the creative defensive tactic of popularity-based password policies, traditionally the purview of system security, into cryptography (see Sec. 4.2). This, thereby, will trigger discussions about the important implications that the progresses in password research (e.g., security, usability and management) would have for the areas of password-based cryptography (e.g., authentication, encryption and signature).

APPENDIX

A.1. Finding and Fixing an inherent flaw in the strength conversion of α-guesswork

To overcome the various problems (e.g., incomparability, inaccuracy and un-repeatability) in existing password strength metrics, Bonneau [2012b] proposed the α-guesswork that relies on the statistical distribution of passwords and is parameterized on an attacker's desired success rate α. It well captures the reality that a practical attacker $\mathcal{A}$ is generally satisfied with cracking the weak fraction of accounts. This metric has been widely used in academia [Chatterjee et al. 2015; Li et al. 2014; Bailey et al. 2014] and also won the NSA 2013 annual award for "Science of Security Competition" [NSA Press Release 2013]. Here we report an inherent flaw in its strength conversion and further manage to fix it.

For better comprehension, here we follow the notations in [Bonneau 2012b] as closely as possible. The probability distribution is denoted by $\mathcal{X}$, each password x_i is randomly drawn from $\mathcal{X}$ with a probability p_i, such that $\sum p_i$=1. Without loss of generality, assume $p_1 \geq p_2 \geq \cdots \geq p_{\mathcal{N}}$, where $\mathcal{N}$ is the the total number of possible events in $\mathcal{X}$. For 0<α≤1, $\mu_\alpha(\mathcal{X})$ = min $\{j | \sum_{i=1}^{j} p_i \geq \alpha\}$ measures the minimal number of fixed guesses per account that $\mathcal{A}$ needs to crack at least a fraction α of total passwords, and $\lambda_\beta(\mathcal{X}) = \sum_{i=1}^{\beta} p_i$ denotes the expected success for $\mathcal{A}$ limited to β guesses per account. Thus, λ_{μ_α} measures $\mathcal{A}$'s actual success when given μ_α guesses per account and $\lambda_{\mu_\alpha} \geq \alpha$. With these terminologies, α-guesswork is defined as:

$$G_\alpha(\mathcal{X}) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i, \tag{19}$$

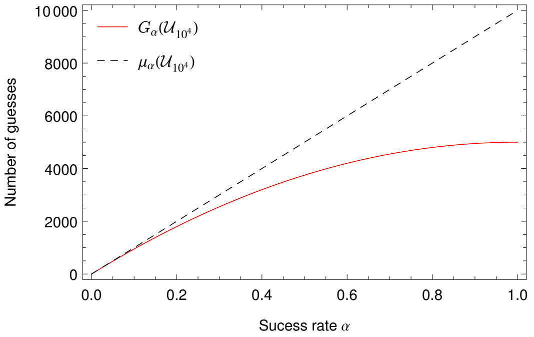


Fig. 9: How $G_\alpha(\mathcal{U}_N)$ and $\mu_\alpha(\mathcal{U}_N)$ vary with α

$G_\alpha(\mathcal{X})$ characterizes the expected number of guesses per account to reach a success rate α. The intuition of Eq.19 is that: (1) against every account not in $\mathcal{A}$'s dictionary she will make μ_α guesses, giving rise to the first term; and (2) against all accounts that are in $\mathcal{A}$'s dictionary, she proceeds in optimal order and the expected number of guesses required constitutes the second term. $G_\alpha(\mathcal{X})$ well models the reality of real-world attackers, who care about cost-effectiveness, to stop cracking against the most strong accounts.

For easier comparison with other existing metrics and for better comprehension of programmers and cryptographers, Bonneau [2012b] further converted $G_\alpha(\mathcal{X})$ into units of bits (i.e., $\widetilde{G}_\alpha(\mathcal{X})$) by computing "the logarithmic size of a discrete uniform distribution $\mathcal{U}_{\mathcal{N}}$ (with $p_i = 1/\mathcal{N}$ for all $1 \leq i \leq \mathcal{N}$) that has the same value of the guessing metric". Since an attacker $\mathcal{A}$ who desires to break a proportion α of accounts will "attain one successful guess per G_α/α guesses", $\mathcal{A}$ will "break an account every $(\mathcal{N}+1)/2$ guesses" against $\mathcal{U}_{\mathcal{N}}$. This gives the formula (see pp.49 of [Bonneau 2012a] for a more detailed explanation):

$$\frac{G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} = \frac{\mathcal{N}+1}{2} \quad \Rightarrow \quad \widetilde{G}_\alpha(\mathcal{X}) = \lg \mathcal{N} = \lg\Big[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\Big] \tag{20}$$

$\widetilde{G}_\alpha(\mathcal{X})$ should have been constant for any uniform distribution $\mathcal{U}_{\mathcal{N}}$, but Bonneau [2012b] found it was not the case. So, he added the "correction factor" $\lg \frac{1}{2-\lambda_{\mu_\alpha}}$ to $\widetilde{G}_\alpha(\mathcal{X})$, giving:

$$\widetilde{G}_\alpha(\mathcal{X}) = \lg\Big[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\Big] + \lg \frac{1}{2 - \lambda_{\mu_\alpha}} \tag{21}$$

However, we will demonstrate that the equation on the left side of Eq.20 is inherently flawed. As can be seen from Fig.2(a) in [Bonneau 2012b], it was believed that $G_\alpha(\mathcal{U}_{\mathcal{N}}) = \mu_\alpha(\mathcal{U}_{\mathcal{N}})$. Quite the contrary, our Fig. 7 well serves as a concrete counter-example that $G_\alpha(\mathcal{U}_{10^4}) \neq \mu_\alpha(\mathcal{U}_{10^4})$. Essentially, according to Eq.19, one can get

$$G_\alpha(\mathcal{U}_\mathcal{N}) = \sum_{i=1}^{\mu_\alpha} i \cdot \frac{1}{\mathcal{N}} + (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha = \frac{(1+\mu_\alpha)\mu_\alpha}{2} \cdot \frac{1}{\mathcal{N}} + (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha \tag{22}$$

On the other hand, according to the definition of μ_α and λ_β in [Bonneau 2012b], we get

$$\mu_\alpha(\mathcal{U}_\mathcal{N}) = \mathcal{N} \cdot \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}) \tag{23}$$

Based on Eq.23, Eq.22 can be rewritten as

$$G_\alpha(\mathcal{U}_\mathcal{N}) = \frac{(1+\mathcal{N}\cdot\lambda_{\mu_\alpha})\cdot\mathcal{N}\lambda_{\mu_\alpha}}{2\mathcal{N}} + (1-\lambda_{\mu_\alpha})\cdot\mathcal{N}\cdot\lambda_{\mu_\alpha} = \frac{\lambda_{\mu_\alpha}}{2} + \frac{1}{2}(2-\lambda_{\mu_\alpha})\cdot\mathcal{N}\cdot\lambda_{\mu_\alpha} \tag{24}$$

From Eq.23 and Eq.24, it is evident that $G_\alpha(\mathcal{U}_\mathcal{N}) \neq \mu_\alpha(\mathcal{U}_\mathcal{N})$. Based on Eq.24, for $\mathcal{U}_\mathcal{N}$ and $\mathcal{X}$ to be of equivalent security, we get

$$G_\alpha(\mathcal{U}_\mathcal{N}) = G_\alpha(\mathcal{X}) \overset{\text{Eq.23}}{\Longrightarrow} \mathcal{N} = \frac{2G_\alpha(\mathcal{X}) - \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})}{(2-\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}))\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})} \tag{25}$$

Note that, for $0<\alpha\leq 1$, $0 \leq \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}) - \alpha < \frac{1}{\mathcal{N}}$ and $0 \leq \lambda_{\mu_\alpha}(\mathcal{X}) - \alpha < p_n$, where $p_n \leq p_{n-1} \leq \cdots \leq p_1$ and $\sum_{i=1}^{n-1} p_i < \alpha \leq \sum_{i=1}^{n} p_i = \lambda_{\mu_\alpha}$. This suggests that $-\frac{1}{\mathcal{N}} \leq \lambda_{\mu_\alpha}(\mathcal{X}) - \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}) \leq q_n$, giving $|\lambda_{\mu_\alpha}(\mathcal{X}) - \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})| \leq \max\{\frac{1}{\mathcal{N}}, q_n\}$. Note that, only when α is large enough (0.5 as a benchmark recommended in [Bonneau 2012b]), $\widetilde{G}_\alpha(\mathcal{X})$ will show advantage over $\widetilde{\mu}_\alpha(\mathcal{X})$; q_n decreases as α increases. When $\alpha \geq 0.2$, $q_n < \frac{1}{1000}$ holds for all our 12 datasets. Further, for human-generated passwords, generally $\mathcal{N} \geq 2^{15}$ [Bonneau 2012b; Li et al. 2014]. All this gives the relationship that, when α is large enough, $\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}) \approx \alpha \approx \lambda_{\mu_\alpha}(\mathcal{X})$. Consequently, both $\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})$ and $\lambda_{\mu_\alpha}(\mathcal{X})$ can be unified as λ_{μ_α}. This for the first time explains why λ_{μ_α} in the equations (10) and (11) of [Bonneau 2012b] leave out the distribution $\mathcal{X}$ or $\mathcal{U}_\mathcal{N}$. Based on this observation and Eq.25, for $0 < \alpha < 1$, the equation on the left side of Eq.20 can be shown to be incorrect:

$$\frac{G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} = \frac{1+\mathcal{N}\cdot(2-\lambda_{\mu_\alpha})}{2} \neq \frac{\mathcal{N}+1}{2} \tag{26}$$

Only when $\alpha = 1$, because $1 = \alpha \leq \lambda_{\mu_\alpha} \leq 1$, λ_{μ_α} will be equal to 1 and $\frac{G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} = \frac{\mathcal{N}+1}{2}$.

Further, using Eq.25, the "effective key-length" (i.e., bit-strength) of $G_\alpha(\mathcal{X})$ can be naturally formulated as

$$\widetilde{G}_\alpha(\mathcal{X}) = \lg\mathcal{N} = \lg\frac{2G_\alpha(\mathcal{X}) - \lambda_{\mu_\alpha}}{(2-\lambda_{\mu_\alpha})\lambda_{\mu_\alpha}} = \lg\Big[\frac{2\cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\Big] + \lg\frac{1}{2-\lambda_{\mu_\alpha}} \tag{27}$$

It follows that there is no need to add a factitious "correction factor" in the strength conversion of α-guesswork, thereby demonstrating and fixing the flaws in [Bonneau 2012a; Bonneau 2012b]. While the effective key-length metric $\widetilde{G}_\alpha(\mathcal{X})$ is overwhelmingly favored over $G_\alpha(\mathcal{X})$ (e.g., [Chatterjee et al. 2015; Song et al. 2015; Bailey et al. 2014; Li et al. 2014]) and it is widely hold that $G_\alpha(\mathcal{U}_\mathcal{N}) = \mu_\alpha(\mathcal{U}_\mathcal{N})$, our above contribution lies not only in identifying and fixing an inherent flaw in the derivation of $\widetilde{G}_\alpha(\mathcal{X})$, but also, equally importantly, in revealing a counter-intuitive relationship: $G_\alpha(\mathcal{U}_\mathcal{N}) \neq \mu_\alpha(\mathcal{U}_\mathcal{N})$.

REFERENCES

Michel Abdalla, Fabrice Benhamouda, and Philip MacKenzie. 2015a. Security of the J-PAKE Password-Authenticated Key Exchange Protocol. In *Proc. IEEE S&P 2015*. IEEE, 1–17.

Michel Abdalla, Fabrice Benhamouda, and David Pointcheval. 2015b. Public-Key Encryption Indistinguishable Under Plaintext-Checkable Attacks. In *PKC 2015*, J. Katz (Ed.). LNCS, Vol. 9020. Springer, 332–352.

Lada A. Adamic. 2014. *Zipf, Power-laws, and Pareto - a ranking tutorial*. http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html.

Anne Adams and Martina Angela Sasse. 1999. Users are not the enemy. *Commun. ACM* 42, 12 (1999), 40–46.

C. Allan. 2009. *32 million Rockyou passwords stolen*. http://www.hardwareheaven.com/news.php?newsid=526.

Mansour Alsaleh, Mohammad Mannan, and P Van Oorschot. 2012. Revisiting defenses against large-scale online password guessing attacks. *IEEE Trans. Dependable and Secure Computing* 9, 1 (2012), 128–141.

Robert L Axtell. 2001. Zipf distribution of US firm sizes. *Science* 293, 5536 (2001), 1818–1820.

Daniel V Bailey, Markus Dürmuth, and Christof Paar. 2014. Statistics on Password Re-use and Adaptive Strength for Financial Accounts. In *Proc. SCN 2014*. LNCS, Vol. 8642. Springer, 218–235.

Mihir Bellare. 1999. Practice-Oriented Provable-Security. In *ISC 1999*, IvanBjerre Damgard (Ed.). LNCS, Vol. 1561. Springer Berlin/Heidelberg, 1–15.

Mihir Bellare, David Pointcheval, and Phillip Rogaway. 2000. Authenticated Key Exchange Secure against Dictionary Attacks. In *Proc. EUROCRYPT 2000*, Bart Preneel (Ed.). LNCS, Vol. 1807. Springer, 139–155.

Fabrice Benhamouda, Olivier Blazy, Cline Chevalier, David Pointcheval, and Damien Vergnaud. 2013. New Techniques for SPHFs and Efficient One-Round PAKE Protocols. In *CRYPTO 2013*, Ran Canetti and JuanA. Garay (Eds.). LNCS, Vol. 8042. Springer Berlin/Heidelberg, 449–475.

F. Bergadano, B. Crispo, and G. Ruffo. 1998. High dictionary compression for proactive password checking. *ACM Trans. Inform. Syst. Secur.* 1, 1 (1998), 3–25.

Matt Bishop and Daniel V Klein. 1995. Improving system security via proactive password checking. *Computers & Security* 14, 3 (1995), 233–249.

J. Bonneau. 2012a. *Guessing human-chosen secrets*. Ph.D. Dissertation. University of Cambridge.

J. Bonneau. 2012b. The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords. In *Proc. 33th IEEE Symp. on Security and Privacy*. IEEE, 538–552.

J. Bonneau, C. Herley, P. Oorschot, and F. Stajano. 2012. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proc. IEEE S&P 2012*. IEEE, 553–567.

Ron Bowes. 2011. *Password dictionaries*. https://wiki.skullsecurity.org/Passwords.

Alan S Brown, Elisabeth Bracken, Sandy Zoccoli, and King Douglas. 2004. Generating and remembering passwords. *Applied Cognitive Psychology* 18, 6 (2004), 641–651.

W. Burr, D. Dodson, R. Perlner, W. Polk, S. Gupta, and E. Nabbus. April 2006. *NIST SP800-63 – Electronic Authentication Guideline*. Technical Report. NIST, Reston, VA.

W. Burr, D. Dodson, R. Perlner, W. Polk, S. Gupta, and E. Nabbus. Aug. 2013. *NIST SP800-63-2 – Electronic Authentication Guideline*. Technical Report. NIST, Reston, VA.

Ran Canetti, Dana Dachman-Soled, Vinod Vaikuntanathan, and Hoeteck Wee. 2012. Efficient password authenticated key exchange via oblivious transfer. In *Proc. PKC 2012*, Marc Fischlin, Johannes Buchmann, and Mark Manulis (Eds.). LNCS, Vol. 7293. Springer Berlin/Heidelberg, 449–466.

Claude Castelluccia, Markus Dürmuth, and Daniele Perito. 2012. Adaptive password-strength meters from Markov models. In *Proc. NDSS 2012*. 1–15.

Rahul Chatterjee, Joseph Bonneau, Ari Juels, and Thomas Ristenpart. 2015. Cracking-Resistant Password Vaults using Natural Language Encoders. In *Proc. IEEE S&P 2015*. IEEE, 1–18.

Liqun Chen, Hoon Wei Lim, and Guomin Yang. 2014. Cross-domain password-based authenticated key exchange revisited. *ACM Trans. Inform. Syst. Secur.* 16, 4 (2014), 15.

Sonia Chiasson and Paul C van Oorschot. 2015. Quantifying the Security Advantage of Password Expiration Policies. *Designs, Codes and Cryptography* (2015). Doi:10.1007/s10623-015-0071-9.

Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

Lucian Constantin. 2009. *Security Gurus 0wned by Black Hats*. http://news.softpedia.com/news/Security-Gurus-0wned-by-Black-Hats-117934.shtml.

Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and XiaoFeng Wang. 2014. The Tangled Web of Password Reuse. In *Proc. NDSS 2014*. 1–15.

X de Carné de Carnavalet and Mohammad Mannan. 2014. From very weak to very strong: Analyzing password-strength meters. In *Proc. NDSS 2014*. Internet Society, 1–16. http://dx.doi.org/10.14722/ndss.2014.23268.

M. Dell'Amico, P. Michiardi, and Y. Roudier. 2010. Password strength: an empirical analysis. In *Proc. INFOCOM 2010*. IEEE Communications Society, 983–991.

S. Designer. 1996. *John the Ripper password cracker*. http://www.openwall.com/john/.

M. Dürmuth. 2013. Useful password hashing: how to waste computing cycles with style. In *Proc. of the 22th workshop on New Security Paradigms Workshop (NSPW 2013)*. ACM, 31–40.

Markus Dürmuth, David Freeman, and Battista Biggio. 2016. Who are you? A statistical approach to measuring user authenticity. In *NDSS 2016*. 1–15.

S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley. 2013. Does my password go up to eleven?: the impact of password meters on password selection. In *Proc. of CHI 2013*. ACM, 2379–2388.

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On Power-law Relationships of the Internet Topology. In *Proc. SIGCOMM 1999*. ACM, New York, NY, USA, 251–262.

Dinei Florencio and Cormac Herley. 2007. A large-scale study of web password habits. In *WWW 2007*. ACM, New York, NY, USA, 657–666.

Dinei Florêncio and Cormac Herley. 2010. Where do security policies come from?. In *Proc. of the Sixth Symposium on Usable Privacy and Security (SOUPS 2010)*. ACM, Redmond, Washington, USA, 1–14.

Dinei Florêncio, Cormac Herley, and Baris Coskun. 2007. Do strong web passwords accomplish anything? *Proc. HotSec 2007* 7 (2007), 6.

Dinei Florêncio, Cormac Herley, and P van Oorschot. 2014. An Administrators Guide to Internet Password Research. In *Proc. USENIX LISA 2014*. 35–52.

K. Gjosteen and O. Thuen. 2012. Password-Based Signatures. In *EuroPKI 2011*. LNCS, Vol. 7163. 17–33.

S. Halevi and H. Krawczyk. 1999. Public-key cryptography and password protocols. *ACM Trans. Inform. Syst. Secur.* 2, 3 (1999), 230–268.

Cormac Herley. 2013. When does Targeting make sense for an attacker? *IEEE Secur. & Priv.* 11, 2 (2013), 89–92.

Cormac Herley and Paul Van Oorschot. 2012. A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy* 10, 1 (2012), 28–36.

Shiva Houshmand and Sudhir Aggarwal. 2012. Building better passwords using probabilistic techniques. In *Proc. ACSAC 2012*. ACM, 109–118.

Xinyi Huang, Yang Xiang, Elisa Bertino, Jianying Zhou, and Li Xu. 2014. Robust Multi-Factor Authentication for Fragile Communications. *IEEE Trans. Depend. Secur. Comput.* 11, 6 (2014), 568–581.

Zhicong Huang, Erman Ayday, Jacques Fellay, Jean-Pierre Hubaux, and Ari Juels. 2015. GenoGuard: Protecting genomic data against brute-force attacks. In *Proc. IEEE S&P 2015*. IEEE, 1–16.

Philip G Inglesant and M Angela Sasse. 2010. 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, 383–392.

Stanislaw Jarecki, Aggelos Kiayias, and Hugo Krawczyk. 2014. Round-Optimal Password-Protected Secret Sharing and T-PAKE in the Password-Only Model. In *ASIACRYPT 2014*, Palash Sarkar and Tetsu Iwata (Eds.). LNCS, Vol. 8874. Springer, 233–253.

Casey Johnston. 2013. *Why your password cant have symbols*. http://arstechnica.com/security/2013/04/why-your-password-cant-have-symbols-or-be-longer-than-16-characters/.

Ari Juels and Thomas Ristenpart. 2014. Honey encryption: Security beyond the brute-force bound. In *Proc. EUROCRYPT 2014*. Springer, 293–310.

Vladimir Katalov. 2013. *Yahoo!, Dropbox and Battle.net Hacked: Stopping the Chain Reaction*. http://blog.crackpassword.com/2013/02/yahoo-dropbox-and-battle-net-hacked-stopping-the-chain-reaction/.

Jonathan Katz, Rafail Ostrovsky, and Moti Yung. 2009. Efficient and secure authenticated key exchange using weak passwords. *J. ACM* 57, 1 (2009), 1–41.

Jonathan Katz and Vinod Vaikuntanathan. 2013. Round-optimal password-based authenticated key exchange. *Journal of Cryptology* 26, 4 (2013), 714–743.

Patrick Gage Kelley, Saranga Komanduri, Michelle L Mazurek, Richard Shay, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, and Julio Lopez. 2012. Guess again (and again and again): Measuring password strength by simulating password-cracking algorithms. In *Proc. IEEE S&P 2012*. IEEE, 523–537.

Daniel V Klein. 1990. Foiling the cracker: A survey of, and improvements to, password security. In *Proc. of the 2nd USENIX Security Workshop*. 5–14.

Saranga Komanduri, Richard Shay, Lorrie Faith Cranor, Cormac Herley, and Stuart Schechter. 2014. Telepathwords: Preventing weak passwords by reading users minds. In *Proc. USENIX SEC 2014*. 591–606.

Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. 2006. Human selection of mnemonic phrase-based passwords. In *Proc. SOUPS 2006*. ACM, 67–78.

Zhigong Li, Weili Han, and Wenyuan Xu. 2014. A Large-Scale Empirical Analysis on Chinese Web Passwords. In *Proc. USENIX Security 2014*. 559–574.

J. Long. 2011. *No tech hacking: A guide to social engineering, dumpster diving, and shoulder surfing*. Syngress.

Jerry Ma, Weining Yang, Min Luo, and Ninghui Li. 2014. A Study of Probabilistic Password Models. In *Proc. IEEE S&P 2014*. IEEE, 689–704.

T. Maillart, D. Sornette, S. Spaeth, and G. Von Krogh. 2008. Empirical tests of Zipf's law mechanism in open source Linux distribution. *Physical Review Letters* 101, 21 (2008), 218701.

D. Malone and K. Maher. 2012. Investigating the distribution of password choices. In *Proc. WWW 2012*. 301–310.

Rick Martin. 2012. *Amid Widespread Data Breaches in China*. https://sg.finance.yahoo.com/news/Amid-Widespread-Data-Breaches-pennolson-706259476.html.

M. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. Cranor, P. Kelley, R. Shay, and B. Ur. 2013. Measuring password guessability for an entire university. In *Proc. CCS 2013*. ACM, 173–186.

Declan McCullagh. 2013. *Feds tell Web firms to turn over user account passwords*. http://www.cnet.com/news/feds-tell-web-firms-to-turn-over-user-account-passwords.

J. Mick. 2014. *Russian Hackers Compile List of 10M+ Stolen Gmail, Yandex, Mailru*. http://www.dailytech.com/Russian+Hackers+Compile+List+of+10+Million+Stolen+Gmail+Yandex+Mailru/article36537.htm.

R. Morris and K. Thompson. 1979. Password security: A case history. *Commun. ACM* 22, 11 (1979), 594–597.

Arvind Narayanan and Vitaly Shmatikov. 2005. Fast dictionary attacks on passwords using time-space tradeoff. In *Proc. CCS 2005*. ACM, 364–372.

M. Newman. 2005. Power laws, Pareto distributions and Zipf's law. *Contemporary Physics* 46, 5 (2005), 323–351.

NSA Press Release 2013. *1st Annual Best Scientific Cybersecurity Paper Competition*. http://cps-vo.org/group/sos/papercompetition2012.

Philippe Oechslin. 2003. Making a Faster Cryptanalytic Time-Memory Trade-Off. In *Proc. CRYPTO 2003*, Dan Boneh (Ed.). LNCS, Vol. 2729. Springer/Berlin Heidelberg, 617–630.

Outpost9.com's Lab 2014. *Word lists*. Outpost9.com's Lab. http://www.outpost9.com/files/WordLists.html.

Peter Paule and Carsten Schneider. 2003. Computer proofs of a new family of harmonic number identities. *Advances in Applied Mathematics* 31, 2 (2003), 359–378.

David Pointcheval. 2012. Password-Based Authenticated Key Exchange. In *Proc. PKC 2012*, Marc Fischlin, Johannes Buchmann, and Mark Manulis (Eds.). LNCS, Vol. 7293. Springer/Berlin Heidelberg, 390–397.

Ashwini Rao, Birendra Jha, and Gananand Kini. 2013. Effect of grammar on security of long passwords. In *Proc. of the 3rd ACM Conf. on Data and Application Security and Privacy (CODASPY 2013)*. ACM, 317–324.

Stuart Schechter, Cormac Herley, and Michael Mitzenmacher. 2010. Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks. In *Proc. HotSec 2010*. 1–8.

R. Shay, P. Kelley, S. Komanduri, M. Mazurek, B. Ur, T. Vidas, L. Bauer, and L. Cranor. 2012. Correct horse battery staple: Exploring the usability of system-assigned passphrases. In *Proc. SOUPS 2012*. ACM, 1–20.

R. Shay, S. Komanduri, P. Kelley, P. Leon, M. Mazurek, L. Bauer, N. Christin, and L. Cranor. 2010. Encountering stronger password requirements: user attitudes and behaviors. In *Proc. SOUPS 2010*. ACM, 1–20.

Y. Song, G. Cho, S. Oh, H. Kim, and J. Huh. 2015. On the Effectiveness of Pattern Lock Strength Meters: Measuring the Strength of Real World Pattern Locks. In *Proc. CHI 2015*. ACM, 2343–2352.

Sophos Press Release 2009. *Security at risk as one third of surfers admit they use the same password for all websites*. Sophos Press Release. https://www.sophos.com/en-us/press-office/press-releases/2009/03/password-security.aspx.

E. Spafford. 1992a. Observations on Reusable Password Choices. In *Proc. USENIX Security Workshop*. 1–14.

E. Spafford. 1992b. Opus: Preventing weak password choices. *Computers & Security* 11, 3 (1992), 273–278.

San-Tsai Sun, Eric Pospisil, Ildar Muslukhov, Nuray Dindar, Kirstie Hawkey, and Konstantin Beznosov. 2013. Investigating Users Perspectives of Web Single Sign-On: Conceptual Gaps and Acceptance Model. *ACM Trans. Internet Tech.* 13, 1 (2013), 1–35.

Blase Ur, Patrick Gage Kelley, Saranga Komanduri, Joel Lee, Michael Maass, Michelle Mazurek, Timothy Passaro, Richard Shay, Timothy Vidas, Lujo Bauer, and others. 2012. How does your password measure up? The effect of strength meters on password creation. In *Proc. USENIX Security 2012*. 65–80.

Paul C Van Oorschot and Stuart Stubblebine. 2006. On countering online dictionary attacks with login histories and humans-in-the-loop. *ACM Trans. Inform. Syst. Secur.* 9, 3 (2006), 235–258.

Ding Wang, Debiao He, Ping Wang, and Chao-Hsien Chu. 2015a. Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals Are Beyond Attainment. *IEEE Trans. Depend. Secur. Comput.* 12, 4 (2015), 428–442.

Ding Wang, Gaopeng Jian, Xinyi Huang, and Ping Wang. 2015b. Supplemental data: Effects of the sample size and the least frequency threshold on linear regression when simulating a given (perfect) Zipf distribution. (Mar. 2015). http://wangdingg.weebly.com/uploads/2/0/3/6/20366987/simulated_zipf.pdf.

Ding Wang and Ping Wang. 2015. The Emperor's New Password Creation Policies: An Evaluation of Leading Web Services and the Effect of Role in Resisting Against Online Guessing. In *Proc. ESORICS 2015*. Springer.

Matt Weir, Sudhir Aggarwal, Michael Collins, and Henry Stern. 2010. Testing metrics for password creation policies by attacking large sets of revealed passwords. In *Proc. CCS 2010*. ACM, 162–175.

Matt Weir, Sudhir Aggarwal, Breno de Medeiros, and Bill Glodek. 2009. Password cracking using probabilistic context-free grammars. In *Proc. 30th IEEE Symp. on Security and Privacy*. IEEE, 391–405.

Jeff Jianxin Yan, Alan F Blackwell, Ross J Anderson, and Alasdair Grant. 2004. Password Memorability and Security: Empirical Results. *IEEE Security & privacy* 2, 5 (2004), 25–31.

Guomin Yang, Duncan S Wong, Huaxiong Wang, and Xiaotie Deng. 2008. Two-factor mutual authentication based on smart cards and passwords. *J. Comput. Syst. Sci.* 74, 7 (2008), 1160–1172.

Xun Yi, Feng Hao, and Elisa Bertino. 2014. ID-Based Two-Server Password-Authenticated Key Exchange. In *Proc. ESORICS 2014*, M. Kutylowski and J. Vaidya (Eds.). LNCS, Vol. 8713. Springer, 257–276.

Ziming Zhao, Gail-Joon Ahn, and Hongxin Hu. 2015. Picture Gesture Authentication: Empirical Analysis, Automated Attacks, and Scheme Evaluation. *ACM Trans. Inform. Syst. Secur.* 17, 4 (2015), 1–37.

B Zhu, Jeff Yan, Guanbo Bao, M Mao, and Ning Xu. 2014. Captcha as Graphical Passwords–A New Security Primitive Based on Hard AI Problems. *IEEE Trans. Inform. Forensics Security* 9, 6 (2014), 891–904.

George Kingsley Zipf. 1949. *Human behavior and the principle of least effort.* Addison-Wesley Press.