Zipf's Law in Passwords

Ding Wang *Student Member, IEEE*, Gaopeng Jian, Xinyi Huang and Ping Wang *Senior Member, IEEE*

Abstract—Despite three decades of intensive research, textual passwords are still enveloped in mysterious veils. It remains an open question as to what is the underlying distribution of user-generated passwords. In this work, we make a substantial step forward towards understanding this question. By introducing a number of computational statistical techniques and based on fourteen large-scale datasets, which consist of 127.7 million real-world passwords, we for the first time show that Zipf's law natively exists in the popular (and thus vulnerable) part of human-generated password datasets. Further, we provide compelling evidence that this law is also highly likely to hold in the remaining part of human-generated passwords. With the concrete knowledge of password distributions, we suggest a new metric for measuring the strength of password datasets. Both theoretical and experimental results show the effectiveness of the proposed metric.

Keywords— Authentication, Password distribution, Zipf's law, Password cracking, Strength metric.

I. Introduction

Password-based authentication is being used for access control by almost every Internet service today. Despite its ubiquity, this kind of authentication is accompanied by the dilemma of generating passwords which are challenging for powerful attackers to crack but easy for common users to remember. It is well known that truly random passwords are difficult for users to memorize, while user-chosen passwords may be highly predictable [1], [2]. In practice, common users tend to gravitate towards weak passwords that are related to their daily lives (e.g., names, birthdays, lovers, friends and hobbies [3], [4]), which means these passwords are drawn from a rather small space and thus are prone to guessing attacks.

To mitigate this notorious security-usability dilemma, various password creation policies have been proposed, e.g., random generation [5], rule-based [6], entropy-based [7] and cracking-based [8]. They force newly created passwords to adhere to some composition rules and to achieve an acceptable strength. The diversity of password rules and strength meters brings about an enormous variety of requirements among different web services, resulting in highly conflicting strength scores for the same password [4]. For example, the password `password$1` is deemed "Very Weak" by Dropbox, "Weak" by Apple, "Fair" by Google and "Very Strong" by Yahoo.

The above contradictory outcomes of password strength (for more concrete examples, see [4], [9]) are a direct result of the inconsistent password strength meters employed among different web services, which may in part be further explained by the un-soundness of current password meters and the diverse interests of each web service. It is a rare piece of good news for the password research community that password policies do impact user password choices and if well-designed, password policies can significantly improve password security while maintaining usability [10]. Accordingly, much attention (e.g., [4], [11], [12]) has been paid to the design and analysis of password policies. While 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 [13]. Results in [14] show that, improper password policies in a specific context of use can increase both mental and cognitive workload on users and impact negatively on user productivity, and ultimately users will try every means to circumvent such un-friendly policies.

As a result, different types of web services typically have quite different favors. For portals like Yahoo! and order accepting sites like Kaspersky, usability is a critical property because anything that undermines user experience may result in loss of users to competitors and impair the success of business. So they tend to have lenient policies [4], [15]. 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 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 symbols and rejection of popular passwords like `pa$$word123`).

As different services favor varied password policies, a number of critical issues arise: how can the policy designers evaluate their policies? How can the administrators select the right policy for their systems? In addition, usually the users of a web service 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 stays the same. This is 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.

Hence, 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 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, here we presume that each existing authentication system has already adopted some password policy (e.g., [8], [16]), and its adjustment mainly involves changing some rules and the password strength threshold.

- *D. Wang, G. Jian and P. Wang are with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. Email: {wangdingg, gpjian, pwang}@pku.edu.cn*
- *X.Y. Huang is with the School of Mathematics and Computer Science, Fujian Normal University, China. Email: xyhuang81@gmail.com*

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 is the distribution that user-generated passwords follow*? Despite more than 30 years of intensive research efforts, 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 hundreds, some recent ones include [17], [18]) still rely on an inconceivable assumption: Passwords follow a uniform distribution.

To the best of our knowledge, the work by Malone and Maher [19] may be the most relevant to what we will discuss in this paper. They for the first time made an attempt to investigate the distribution of passwords. They employed four password datasets (three of which are with a size smaller than 10^5) and reached the conclusion that, their datasets are "unlikely to actually be Zipf distributed".[1] Such a conclusion is right contrary to what we will show in the current work. They 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 used an inherently flawed method to attempt to model password distribution with Zipf (naturally, they failed), and they compared their model with a uniform model, and the comparison results showed 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 no much sense. This confusing, unsatisfactory situation motivates our work.

A. Our contributions

In this work, we bring the understanding of the distribution of real-life passwords and the evaluation of password datasets onto a sound scientific footing by adapting statistical techniques, and make the following key contributions:

A Zipf model. We adopt techniques from computational statistics to show that Zipf's law exists in real-life passwords: (1) the vulnerable portion of user-chosen passwords (i.e., popular passwords such as those with a frequency $f \geq 3$) *natively* follows a Zipf-distribution; and (2) the remaining portion of user-chosen passwords (i.e., un-popular passwords such as ones with a frequency $f \leq 2$) is *highly likely* to follow a Zipf-distribution. Extensive empirical experiments on fourteen large-scale real-world password datasets demonstrate the soundness of our Zipf model. This suggests that each password can be seen as a specific sample drawn from the underlying password population which follows the Zipf's law. This invalidates the claim made in [19], [20] that user passwords are "unlikely to actually be Zipf distributed".

A strength metric. We propose a novel metric for measuring the strength of a given password dataset. This metric 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 [8] and non-deterministic nature in α-guesswork [20]). Our metric facilitates a better grasp of the strength of password datasets (either in plain-text or hashed form) in a mathematically rigorous manner, making it possible for security administrators to precisely evaluate the security property of a password policy under which these password datasets are created.

Some insights. We show an implication of our Zipf theory for how to choose the right threshold of popularity-based password policies (e.g., [6]). We for the first time provide a sound rationale that explicates the necessity and feasibility (as well as precautions) for popularity-based password policies. Besides, we report an inherent flaw in the strength conversion of α-guesswork [20] and manage to figure out how to fix it.

II. Related Work

We now briefly review some related works on password policy and password cracking to facilitate later discussions.

A. Password creation policies

In 1990, Klein proposed the concept of proactive password checker, which enables users to create more secure password distributions and checks, a priori, whether the newly submitted passwords are "safe" [21]. The criteria can be divided into two types. One type is the exact rules for what constitute 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 called the technique "proactive password checking", it is indeed the same as password 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 (e.g., Opus [22]). 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 [7]. However, by modeling the success rates of current password cracking techniques against real-life user passwords created under different rules, Weir et al. [11] 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 [8] proposed a novel policy that improves password security while maintaining usability: it 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 V) is suggested by Schechter et al. [6]. 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

[1] Almost at the same time, Bonneau [20] employed essentially the same approach with [19] and as expected, the same conclusion with was reached in [20]. Thus, we mainly use Malone-Maher's work [19] for discussion.

to online guess passwords is left as 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 [6]), yet whether it would greatly reduce usability has not been evaluated thoroughly (e.g., how easily users may be able to adapt to this new policy? No theoretical or empirical user case study results have ever been 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, as we will show in Appendix B, 34.89% of users in www.tianya.cn use passwords that are more frequent than $\mathcal{T}$=$\frac{1}{10^6}$, which indicates that over 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.

B. Password cracking

Password-based systems are prone to various attacks, such as on-line guessing, offline guessing, keylogging, shoulder surfing and social engineering. Here we only consider the on-line and offline guessing attacks, while other attack vectors are unrelated to password strength or password dataset strength and thus outside the scope of this work. Online guessing can be well thwarted by non-cryptographic techniques, such as modern machine-learning-based rate-limiting or locking strategies, while offline guessing 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.

Florencio et al. [23] 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 into 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 by the victim site only after some period of time (e.g., a few days). During this period, offline password 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 system (e.g., which hash function is used, SHA-1, PBKDF2 or CASH [24]?) 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 [25]. For example, system countermeasures against offline attacks, such as salting to defeat pre-computation techniques (e.g., Rainbow tables) or key strengthening 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 measure is followed in 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 or something related to their daily lives) that are easily rememberable [1], [3]. However, users rarely use unmodified elements from such lists, for instance, because password 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 `pa$$word` is generated by leeting two letters of the easily guessable `password`.

To model this password generation practice, researchers utilize various heuristic mangling rules to produce variants of words from an input dictionary. For some widely used dictionaries, see [26]. This sort of techniques has emerged as early as 1979 in Morris-Thompson's analysis of 3,000 passwords [27]. This initial work has been followed by independent works [21], [28]. Later on, some dedicated software tools like John the Ripper (JTR) [29] appeared. Subsequent studies (e.g., [10], [11]) have often utilized these automated 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 [30] 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 show the effectiveness of their algorithm.

In 2009, on the basis of probabilistic context-free grammars (PCFG), Weir et al. [28] 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, the password `pa$$word123` is denoted by the structure $\mathsf{L}_2\mathsf{S}_2\mathsf{L}_4\mathsf{D}_3$. 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 JTR [29].

In 2014, Ma et al. [31] introduced natural language processing techniques, such as smoothing and normalization into Markov-chain-based password cracking algorithms. They found that, when tuned with the right order and employing some appropriate ways to deal with the problems of data sparsity and normalization, Markov-chain-based cracking algorithms would perform better than PCFG-based cracking algorithms. Therefore, in this work (see Section IV-C) we follow Ma et al.'s Markov-based algorithms to evaluate the collected datastets and make comparisons based on our proposed metric.

In 2015, Ur et al. [32] investigated how the above cracking algorithms used by researchers compare to real-world cracking by professionals and how the choice of cracking algorithms influences research conclusions. They found that each cracking algorithm is highly sensitive to its configuration and that relying on a single cracking approach to evaluate the strength of *a single password* may underestimate the vulnerability to an experienced attacker, while the comparative evaluations of *a password dataset* can rely on a single algorithm.

TABLE I. BASIC INFORMATION ABOUT THE FOURTEEN REAL-LIFE PASSWORD DATASETS

Dataset	Web service	Location	Language	When leaked	How leaked	Total passwords	Unique passwords
Tianya	Social forum	China	Chinese	Dec. 4, 2011	Hacker breached	30,233,633	12,614,676
Dodonew	Gaming&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
000webhost	Web hosting	USA	English	**Oct. 28, 2015**	PHP programming bug	15,251,073	10,583,709
Yahoo	Web portal	USA	English	July 12, 2012	Hacker breached	453,492	342,515
Gmail	Email	Russia	Mainly Russian	Sep. 10, 2014	Phishing&hacking	4,929,090	3,132,028
Mail.ru	Email	Russia	Russian	Sep. 10, 2014	Phishing&malware	4,932,688	2,954,907
Yandex.ru	Search engine	Russia	Russian	Sep. 09, 2014	Phishing&malware	1,261,810	717,203

TABLE II. 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%

III. PRELIMINARIES

In this section, we first describe the collected datasets, and then report some statistics about user-chosen passwords. Finally, we give some background on the statistical techniques used—linear regression and Kolmogorov-Smirnov (KS) test.

A. Description of the password datasets

We have collected fourteen large-scale real-life password lists (see Table I) over a time span of nearly ten years. They are different in terms of service, size, how leaked, user localization, language, faith and culture background, suggesting that our model is a generic one and can be used to well characterize the distribution of user-chosen passwords. All fourteen datasets were compromised by hackers or leaked by anonymous insiders, and were subsequently disclosed publicly on the Internet. Some early ones of them have also been used by a number of scientific works that study passwords (e.g., [11], [31], [32]). 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.

The first four datasets, namely Tianya, Dodonew, CSDN and Duowan, are all from Chinese web services. 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 [33] and we collected them at that time. CSDN is the largest community website of Chinese programmers; Tianya is one of the most influential Chinese BBS; Duowan is a popular game forum; Dodonew is also a popular game forum and it enables monetary transactions. 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 [26] which contained 41,545 plain text passwords. The following two datasets are the "Singles.org" and the "Faithwriters". They are both 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) [34]. 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 V). 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 six datasets (i.e., Rockyou, 000webhost, Yahoo, Gmail, Yandex.ru and Mail.ru) to show the generalizability of our findings of Zipf's law in Section IV, and due to space constraints, they will not be analyzed elsewhere. The Rockyou dataset includes

TABLE III. 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%

*Note that the first row is written in regular expressions. For instance, `[a-z]+` means passwords composed of *only* lower-case letters; `[A-Za-z]+` means passwords composed of *only* letters; `[a-zA-z]+[0-9]+` means passwords composed of letters, followed by digits.

TABLE IV. 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%

32M passwords leaked from the gaming forum Rockyou in Dec. 2009 [35]; The 13M 000webhost passwords was made online by the hackers in Oct. 2015, and the 000Webhost officials confirmed the breach and said it was the result of hackers who exploited an old version of the PHP programming language [36]; The 450K Yahoo passwords was made online by the hacker group named D33Ds in July 2012; The last three deatasets (i.e., 4.9M Gmail, 4.9M Mail.ru and 1.3M Yandex.ru) were leaked by Russian hackers in Sep. 2014, and about 90% of them are active [37], 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).

B. Statistics about user-chosen passwords

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 II, `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., [8], [16]) 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.

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 top-3 of the 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 dataset is still very concentrated and among the weakest ones (see Section V). This means that seemingly complex passwords may not be difficult to crack and actually may be rather weak, which further suggests the necessity of a foundational understanding of passwords.

The character composition information is summarized in Table III. Chinese users are more likely to use only digits to construct their passwords, while English users prefer using letters. This complies with [38]. A plausible explanation may be that Chinese users, who usually use hieroglyphics, are less familiar with English words and letters. It is interesting to see that, Myspace users tend to build their passwords by adding the digit "1" to a sequence of lower-case letters. This may be due to its policy that passwords shall include at least one digit.

Table IV shows the length distributions of each dataset. We can see that the most popular password lengths are between 6 and 10, which on average account for 85.01% of the whole dataset. Few users choose passwords that are longer than 12, with Dodonew being an exception. One telling 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 web services) started with a loose password policy and later on enforced a strict policy (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 [39], and a plausible reason may be that the shift to longer allowed password lengths is a non-trivial issue.

C. Linear regression

In statistics, linear regression is the most widely used 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 $R^2 \in [0, 1]$) 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.

D. The Kolmogorov-Smirnov test

Besides R^2, we further employ statistical tests to measure the "distance" between the sample and the theoretic distribution model. Since passwords are unlikely to obey the normal distribution, non-parametric tests shall be used. KS test is one of the most popular non-parametric tests for discrete data [40], [41]. It quantifies the distance between the cumulative distribution function (CDF) $F_n(x)$ of an empirical distribution and the CDF $F(x)$ of the theoretic distribution:

$$D = \sup_x |F_n(x) - F(x)|,$$

where n is the sample size and $\sup_x$ is the supremum of the set of distances. This statistic D can be adopted to conduct a rigorous test. Since the CDF of D is given by

$$\Pr(\sqrt{n}D \leq x) = 1 - 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2x^2} = \frac{\sqrt{2\pi}}{x}\sum_{i=1}^{\infty}e^{\frac{(2i-1)^2\pi^2}{-8x^2}},$$

to see how unlikely such a large outcome of D would be if the hypothesis is true, one can compute the p-value by:

$$\Pr(\sqrt{n}D > x) = 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2x^2}.$$

The null hypothesis is that the assumed theoretic distribution is acceptable, while the alternative is that it is not. A larger p-value indicates it is safer for us to assume that the data tested is not significantly different from the hypothesized distribution.

IV. The Zipf's Law in User-Chosen Passwords

We now provide a new observation of passwords and show that, as opposed to previous research (e.g., [19], [20]), Zipf's law is highly likely to exist in real-life passwords. Besides regular statistical tests, we further justify our methodology in a reverse prospective by simulating a perfect Zipf's distribution and seeing its regression behavior. We also show the wide applicability of our Zipf model.

A. Our methodology

Initially, probabilistic context-free grammars (PCFG) is a machine learning technique used in natural language processing (NLP), yet Weir et al. [28] managed to exploit it to automatically build password mangling rules. Very recently, NLP techniques have also been shown useful in evaluating the security impact of semantics on passwords [42] and in dealing with the sparsity problem in passwords [31].

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, and 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. Zipf's law was shown to account remarkably well (i.e., $R^2 \approx 1$) for the distribution of intensity of wars [41], software packages [44] and the Internet topology [45].

Interestingly, by excluding the least popular passwords from each dataset (i.e., passwords with less than three or five counts) and using linear regression, we find 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 probably 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, profession 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 $\log(r)$:

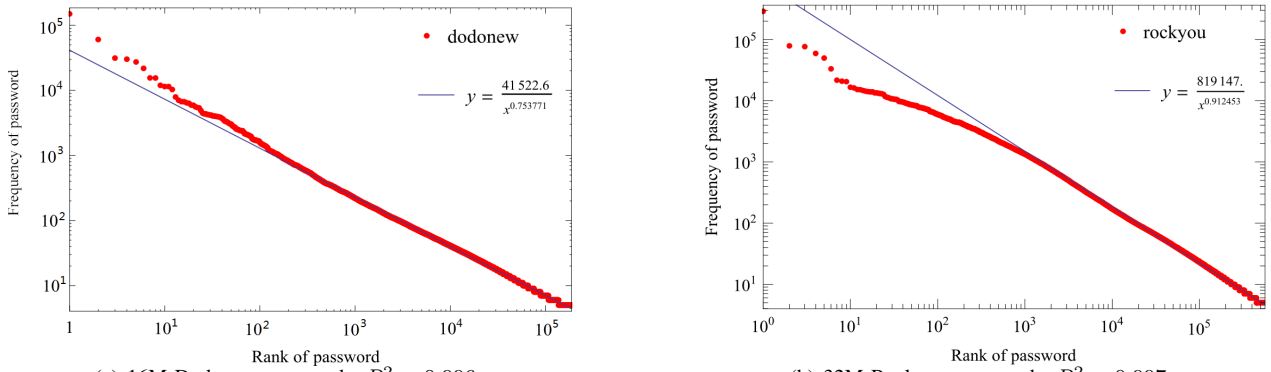
$$\log f_r = \log C - s \cdot \log r. \tag{2}$$

As can be seen from Fig. 1(a), 16.23 million passwords from the website www.dodonew.com conform to Zipf's law to such an extent that the coefficient of determination (denoted by R^2) is 0.995531, which approximately equals 1. This indicates that the regression line $\log y = 4.618284 - 0.753771 * \log x$ well fits the popular passwords from Dodonew. This popular part is the primary security concern as it consists of just these vulnerable passwords: attackers would try these popular passwords first [20]. As illustrated in Fig. 1(b) and Fig. 2, passwords from the other twelve 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 R^2 of 0.923.

[2] Zipf's law distributions are also called Pareto or power-law distributions, and they can be derived from each other when the variable is continuous [43].

TABLE V. LINEAR REGRESSION (LR) RESULTS OF FOURTEEN PASSWORD DATASETS ("PWS" STANDS FOR PASSWORDS)

Dataset	Totoal PWs	Least freq. used	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)	Kolmogorov-Smirnov test	
								Statistic D	p-value
Tianya	30,233,633	5	0.50443286	486,118	0.905773	$5.806523 - 0.905773*\log x$	0.994204954	0.005190	0.075972
Dodonew	16,231,271	5	0.21640911	187,901	0.753771	$4.618284 - 0.753771*\log x$	0.995530686	0.001746	0.412139
CSDN	6,428,287	5	0.29841262	57,715	0.894307	$4.886747 - 0.894307*\log x$	0.985106832	0.001338	0.318784
Duowan	4,982,740	5	0.28653592	51,797	0.841926	$4.666012 - 0.841926*\log x$	0.976258449	0.004455	0.453535
Myspace	41,545	3	0.08094836	706	0.459808	$1.722674 - 0.459808*\log x$	0.965861431	0.000794	0.600451
Singles.org	16,250	3	0.22135384	658	0.518096	$1.875405 - 0.518096*\log x$	0.970277755	0.001452	0.743150
Faithwriters	9,709	3	0.12472963	242	0.486348	$1.583425 - 0.486348*\log x$	0.974175889	0.000376	0.899661
Hak5	2,987	3	0.15400067	76	0.643896	$1.579116 - 0.643896*\log x$	0.922662999	0.009256	0.000019
Rockyou	32,603,388	5	0.49600581	563,074	0.912453	$5.913362 - 0.912453*\log x$	0.997298647	0.004994	0.071003
000webhost	15,251,073	5	0.19687867	229,725	0.624446	$7.354124 - 0.624446*\log x$	0.989437653	0.002056	0.621795
Yahoo	453,492	3	0.22668537	12,608	0.675910	$3.176150 - 0.675910*\log x$	0.983232690	0.002463	0.354603
Gmail	4,926,650	5	0.29617143	77,397	0.799443	$4.903847 - 0.799443*\log x$	0.995817202	0.004374	0.170518
Mail.ru	4,938,663	5	0.33034872	83,914	0.732600	$4.332851 - 0.732599*\log x$	0.970047769	0.004945	0.206273
Yandex.ru	1,261,810	5	0.34210777	26,003	0.620519	$3.394671 - 0.620519*\log x$	0.972507203	0.008792	0.000155



(a) 16M Dodonew passwords: $R^2 = 0.996$ (b) 32M Rockyou passwords: $R^2 = 0.997$

Fig. 1. Zipf's law in two example password datasets. Dodonew includes passwords of Chinese users, while Rockyou includes passwords of English users.

More precisely, as summarized by the "Coefficient of determination" column in Table V, every regression (except for Hack5) is with a R^2>0.965, which closely approaches to 1 and indicates a remarkably sound fitting. As for "Hak5", its R^2 is 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 3000 passwords and probably can not represent the real distribution of the entire password dataset of www.hak5.org. What's more, 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 generally can only obtain a limited portion of a website's passwords, while website breaches, once succeed, all (or at least an overwhelming part of) of the website's passwords will be harvested.

In addition, we employ the KS test [40], [41] to evaluate the goodness-of-fit and 12 out of the 14 regressions are with a p-value>0.05 (see Table V). This means that, at the most widely recognized 0.05 significance level,[3] all these 12 datasets exhibit no statistically significant difference from a Zipf-like distribution. Again, the low KS p-values of Hak5 and Yandex.ru are likely due to the fact that these two datasets are obtained by phishing and thus their representativeness of human behaviors may be insufficient.

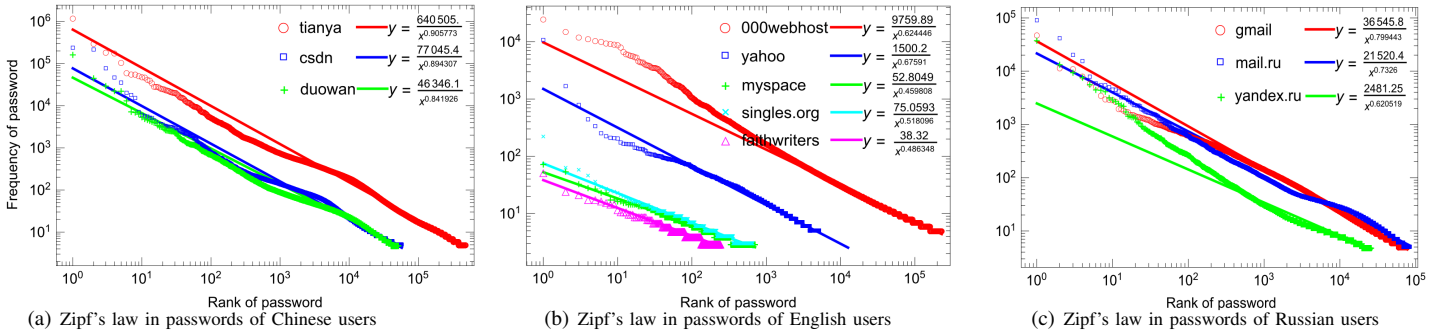
The reason why we need to prune the least frequent passwords will be elaborated in Section IV-B. The selection of a specific small value (e.g., 3 or 5) as the threshold of least frequency (LF) is essentially based on the findings in statistics that (see Fig. 3 of [41]): when *the sample size* is smaller 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 incremental experiments to identify the exact LF that enables the regression to reach $\hat{p}$, and find that, as a useful guideline, for large datasets of million-scale, one can set $LF = 5$, otherwise set $LF = 3$. Note that, to qualify as a proper model for a dataset, a distribution function $f(x)$ shall hold within a range $x_{min} \leq x \leq x_{max}$ of at least $2 \sim 3$ orders of magnitude (i.e., $x_{max}/x_{min} \geq 10^{2\sim3}$) [44]. Except for Hack5, this condition is satisfied by all our regressions.

TABLE VI. COMPARISON OF OUR METHOD WITH THAT OF [41]

Dataset	Zipf model from [41]				Our Zipf model		
	x_{min}	Zipf s=$\frac{1}{\alpha-1}$	KS p-value	Time(sec.)	Zipf s	KS p-value	Time(sec.)
Myspace	3	0.495050	0.012627	1.562	0.459808	0.600451	0.053
Yahoo	4	0.709220	0.167721	166.728	0.675910	0.354603	0.495
Yandex.ru	1	0.543478	0.015413	9515.634	0.620519	0.000155	1.761
Mail.ru	1	0.490196	4.65E-13	82962.997	0.732600	0.206273	5.612

We have also used more complex ways (see pp.12 of [41]) to estimate this threshold and attempted to more accurately determine the distribution parameters, yet these methods are unworkable due to two reasons. Firstly, they first need to determine the parameters of a Power-law distribution, and then convert the Power-law parameters to the Zipf's law parameters. This is unsuitable for discrete variables (e.g., rank of password in our setting). We have tried some conversions (see Table VI), yet the KS tests reject most of them (i.e., p-value<0.05). Secondly, their time complexity is in $\mathcal{O}(|\mathcal{DS}|^2)$, which is unsuitable for modeling large datasets. For instance, when using the codes provided by [41] to process 32M Rockyou, it would take over 306 hours to complete on a moderate computer (i7-4790K 4.00GHz CPU and 16G RAM). In contrast, our

[3]Because of the effect of sample size on the practical significance of a statistical test [46], in order to kept the 0.05 significance level meaningful for our million-sized datasets, we adjust the KS sample size for each dataset to $5*10^5$, a comparable one to [40], [41]. For more details, see Appendix A.



(a) Zipf's law in passwords of Chinese users (b) Zipf's law in passwords of English users (c) Zipf's law in passwords of Russian users

Fig. 2. Zipf's law in eleven real-life password datasets from three different populations, plotted on a log-log scale. Detailed Zipf parameters are referred to Table V. Though a few top-popular passwords do not lie on the fitting line, they are negligible as compared to the ones that dwell on the fitting line.

simple approach is in $\mathcal{O}(|\mathcal{DS}|)$ time complexity and would take only 37.58s. Fortunately, the regression results in Table V show that our selection of the LF threshold is satisfactory: every regression attains a R^2 close to 1 and most of the KS tests accept our Zipf assumption (i.e., p-value>0.05).

Two other critical parameters involved 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 social phenomena (e.g., intensity of wars and frequency of family names [41]) that are with $s > 1$.

B. Justification for our methodology

Malone and Maher [19] 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 [19]) or the "noisy tail" in the statistical domain [43], 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 $\sigma = p_{pw_i}(1 - p_{pw_i})$ [20], 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 [19] 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., 30 million Tianya and 32 million Rockyou) are still *too small to wholly exhibit this intrinsic feature*. For example, csdn.net adopts a policy that allows passwords consisting of letters and numbers and with a length of 8 to 16. This 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 size as compared to $|X|$. Owing to the *polynomially decreasing nature of probability* in a Zipf distribution (see Eq.1), low probability events (e.g., with $f < 3$) 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 (i.e., popular ones) 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) is highly likely to follow 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://t.cn/R4cVxeo) on the distribution of user-chosen PINs, a special kind of passwords, well confirms 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 $f < 10$ are excluded, there are still over 94% of the datasets left in the regression, well following the Zipf's law ($R^2 > 0.97$).

TABLE VII. EFFECTS OF SAMPLE SIZE AND LEAST FREQUENCY (LF) ON REGRESSION WHEN SIMULATING A ZIPF DISTRIBUTION. THE BEST SIMULATIONS ARE IN BOLD.

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

*For the 120 complete experiments, readers are referred to http://t.cn/R4ccgiF.

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 behavior. 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 VII only includes 35 experiments where Zipf N is fixed to 10^3, Zipf s is fixed to 0.9, the sample size varies from 10^2 to 10^4 and LF increases progressively from 1 to 5. Readers are referred to all 120 experimental results in http://t.cn/R4ccgiF. Note that some integral statistics (e.g., the fitted N) in Table VII are with decimals, because they are averaged over 1000 repeated experiments.

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 is, 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., $f \geq 4$) invariably account for over 90% of each sample and well 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 constitute the majority yet we have to exclude them to obtain a good fitting. This justifies our methodology of data processing when performing regression analyses, because the sizes of real-life password datasets are generally much smaller than the password sample space. Overall, the behaviors shown in our regressions on 14 datasets well accord with the 120 simulated experiments, thereby confirming our assertion that the password population is highly likely to follow the Zipf's law.

C. General applicability of the Zipf model

From Section III we can see that, our fourteen datasets include passwords created before 2006 (see Myspace) and also as recent as Oct. 2015 (see 000webhost), cover 12 kinds of web services and three kinds of languages, and represent a variety of culture (faith) backgrounds. Fortunately, both coefficient of determination and KS test show that, these diversified datasets well follow Zipf's law. This to a large extent demonstrates the wide applicability of our Zipf model.

However, in previous regressions we have only focused on datasets that are generated under loose password creation policies. Table II∼IV show that quite short and letter-only passwords appear in every dataset, which suggests that there is no evident length or character requirement for generating passwords in any site. Arguably, a more precise explanation for this phenomenon is that most of these passwords are created under a mixture of unknown policies: Initially, there is no rule; Later on, some stricter (or looser) rule(s) is applied; Sometime later, the sites were hacked. Yet, 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 password 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, it may be meaningless to simply divide an existing dataset according to some artificial policy, because 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" [13]. Mazurek et al. [12] reported a similar observation. Fortunately, after careful examination of our fourteen datasets (see Table III and Table IV), we find that:

(1) Only 2.17% passwords in CSDN are with a length<8. 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 can be confirmed by [4];
(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. 3(a) and 3(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 fourteen 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. 3(c) $\sim$ Fig. 3(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.

To the best of our knowledge, the datasets used in this work are so far the most diversified and among the largest ones, and they are of sound representativeness. It is expected that our Zipf model would provide a much better understanding of the distributions of human-generated passwords and can be widely applicable. With our Zipf theory, now it is becoming possible to compute the right threshold for popularity-based password policies (see Appendix B) and to accurately assess the strength of password datasets as we will show in the following section.

V. 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 is suggested.

A. Our metric

Normally, a smart offline guessing attacker,[4] 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 [20], [25].[5] Accordingly, we can use the cracking result $\lambda^*(n)$ to be the strength metric of a given password dataset:

$$\lambda^*(n) = \frac{1}{|\mathcal{DS}|}\sum_{r=1}^{n} f_r, \tag{3}$$

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

In Section IV, 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):

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

It should be noted that, in Eq. 4, $\lambda^*(n)$ 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. 3 and $\lambda(n)$ as a function of n according to Eq. 4, and put these two curves together to see how they agree with each other. In Fig. 4(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 $\lambda(n)$ $(1 \leq n \leq |\mathcal{DS}|)$ for each dataset in Table VIII.

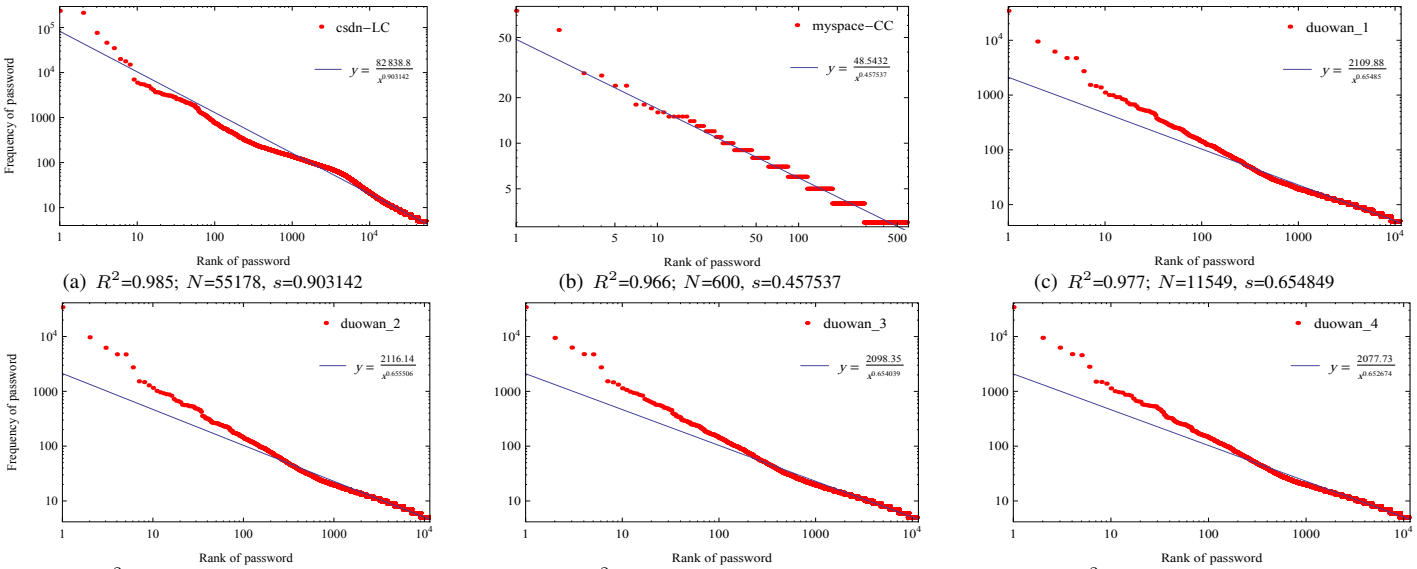
TABLE VIII. The average deviation between $\lambda^*(n)$ and $\lambda(n)$ $(1 \leq n \leq |\mathcal{DS}|)$ for each dataset

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

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. 4, 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

[4]The attacks mentioned in this Section are all offline attacks, because 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 II-B).

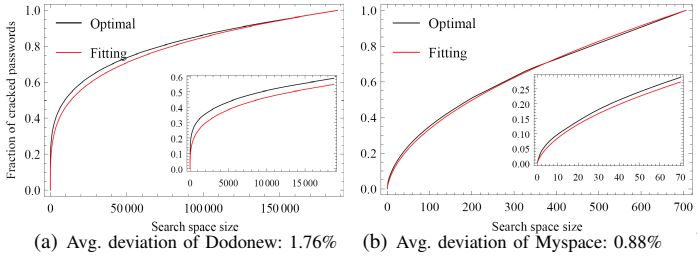
[5]Note that, the optimal attack is of theoretic value (i.e., providing the upper bound) to characterize the best attacking strategy that an attacker can adopt. In practice, if an attacker has already obtained all the plain-text passwords, there is no need for her to order these passwords to crack themselves.



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

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

Fig. 3. Zipf's law in passwords created under constrained policies and in passwords randomly sampled from a real-life dataset (using Duowan as an example).



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

Fig. 4. Consistence of optimal attack with our metric on two example datasets (16.2M Dodonew and 41.5K Mysapce). Our metric performs well.

mainly caused by the deviation of the first *few* high-frequency passwords from the Zipf fitting line (see Fig. 2).

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 service's password dataset becomes weak after a period of time, but also can be used to compare the strength of datasets from different web services. This feature is rather appealing, for the confidence of security only comes after comparison–having a comparison with other similar web services, the security administrators now have a clearer picture about what level of strength their datasets can provide. The recent litany of catastrophic password leakages (e.g., [33], [37]) provides good materials to facilitate such comparisons.

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

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

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

The theorem will be proved in Section V-B, and in Section V-C 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 [31]), respectively.

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

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

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

The above theorem and corollary indicate that, given two password datasets A and B, we can first use liner regression to obtain their fitting lines (i.e., N_A, s_A, N_B and s_B), and then compare N_A with N_B, s_A with s_A, respectively. 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 discrete 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 [28] and markov-based [31]).

The most relevant statistical-based metric to ours may be Bonneau's α-guesswork [20], which has won the NSA 2013 annual "Best Scientific Cybersecurity Paper Award" (see Appendix C). We find this metric is subject to an inherent flaw, and fortunately we manage to fix it. The flaw and the fix do not affect our following analysis however, and thus they are presented in Appendix C. In all four cases, α-guesswork [20] 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 [20] 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 simpler.

Some Remarks. Note that, as with the entropy metric recommended in the NIST SP800-63-2 document [7] and the α-guesswork proposed in [20], 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., [7], [20] and ours). For salted-hash passwords, one needs to resort to attacking-based approaches (e.g., [31]), albeit at the cost of reduced accuracy (as we will show in Section V-C, attacking-based approaches in their current form are subject to too many uncertainties). Also note 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.

B. 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 $1 \le n \le N - 1$,

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

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 N_C=N_A, s_C=s_B, then

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

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

C. 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 is practically effective. It has recently been shown [31], [32] that Markov-based cracking algorithm generally performs better than other ones (e.g., PCFG-based [28] and JTR [29]), and thus we prefer Markov-based algorithm to characterize real-world attacks.

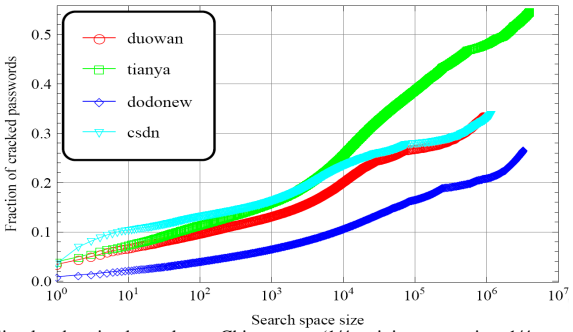
As the optimal attack is of theoretical importance to serve as the ultimate goal of any real attacks, it can by no means be seen as a realistic attack, for it assumes that the attacker is with *all* the plain-text passwords of the target authentication system. To see whether our metric is consistent 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 uses them to guess another quarter of the target system's passwords in any form (salted-hash or unsalted-hash). Note that this new assumption is much more realistic, because most of the compromised web services 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:[6] 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 shown in Fig. 5(a), and results on group two are shown in Fig. 5(b). For any two datasets in the same group, the attacking results comply with our metric results listed in Table V. For instance, from Fig. 5(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.

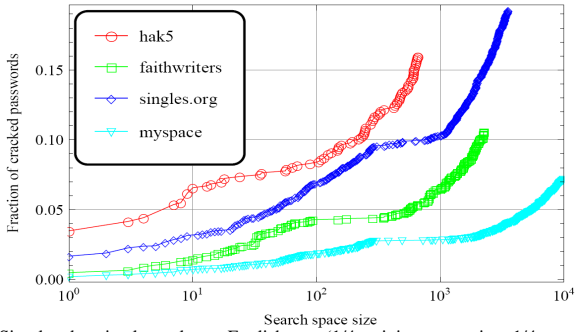
Furthermore, we perform more realistic guessing attacks (i.e. Markov-based attacks) to assess the effectiveness of our metric. As in simulated optimal attacks, we divide the eight main datasets into two groups according to their sizes and languages. For the Chinese group, we use CSDN as the Markov training set; For the English group, we use Myspace as the Markov training set. As shown in [31], 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 of passwords are depicted in Fig. 5(c) and Fig. 5(d), respectively.

It can be seen 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. 5(b) shows that, under the Markov-based guessing attack, Hak5 is the strongest among the three English

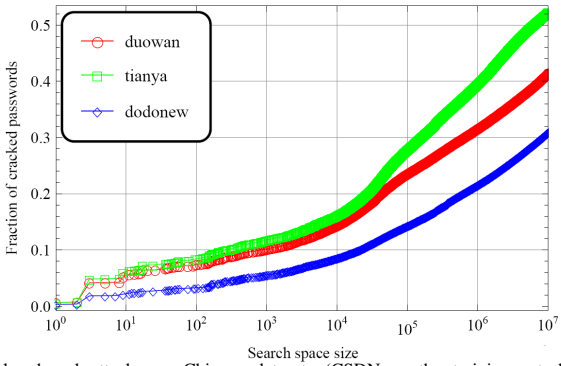
[6] As said earlier, due to space constraints the six auxiliary datasets (see Table I) are only shown to be Zipf-distributed, and actually, all the other general properties revealed in this work are also hold by them.



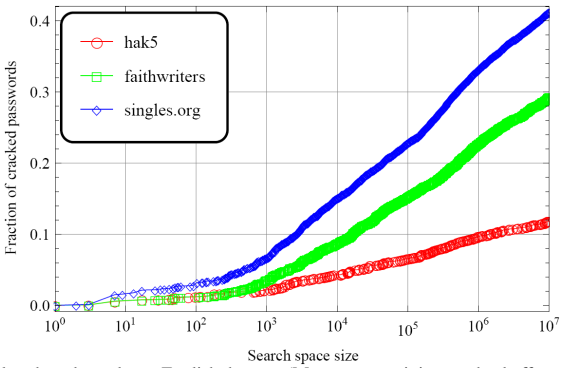
(a) Simulated optimal attacks on Chinese sets (1/4 training set against 1/4 test set)



(b) Simulated optimal attacks 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. 5. Simulated optimal attacks (see Figs. 5(a)∼5(b)) and state-of-the-art real-world attacks (see Figs. 5(c)∼5(d)) on two groups of datasets.

test sets. This inconsistence may be because of its non-representative nature of real-world human password behaviors , or due to the inappropriateness of the selected training set and parameters for the Markov-based cracking algorithm.

Of particular interest may be our observation that, in some cases, 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 only achieve success rates 14.5%∼28.1%, quite 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 heavily on the choices of training sets, smoothing/normalization techniques and may also 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 indicates that there is still room for developing more practical attacking algorithms that have fewer uncertainties yet are more effective. In a nutshell, this highlights the intrinsic limitations of using empirical attacking results (e.g., [8], [16]) as the strength measurement of a password dataset, suggesting the necessity of our metric.

VI. Conclusion

In this work, we have provided compelling answers to the fundamental questions: (1) *What is the underlying distribution of user-generated passwords?* and (2) *How to accurately measure the security strength of a given password dataset?* More specifically, by adopting techniques from computational statistics and using 14 real-life large-scale datasets of 127.7 million passwords, we show that Zipf's law well describes the skewed distributions of passwords; by exploiting the concrete distribution function of passwords, we propose a new statistic-based metric for measuring the strength of a given password dataset, and both theoretical and empirical evidence establish the soundness of our metric. It is expected that the unveiling of Zipf's law in passwords is also of interest in other password research domains, and this work lays the foundation for their further theoretical development and practical application (e.g., the recent "GenoGuard" password cryptosystem [47] and "CASH" password hash function [24] have employed both the theoretical law and numerical results presented in this work).

More work remains to be done on this interesting yet challenging topic. For instance, 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 evolve as time goes on? Do extremely high value passwords (e.g., for e-banking accounts) 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 password datasets are made publicly available.

References

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

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

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

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

[5] J. H. Huh, S. Oh, H. Kim, K. Beznosov, A. Mohan, and S. R. Rajagopalan, "Surpass: System-initiated user-replaceable passwords," in *Proc. CCS 2015*, pp. 170–181.

[6] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks," in *Proc. HotSec 2010*, pp. 1–8.

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

[8] S. Houshmand and S. Aggarwal, "Building better passwords using probabilistic techniques," in *Proc. ACSAC 2012*, pp. 109–118.

[9] X. Carnavalet and M. Mannan, "From very weak to very strong: Analyzing password-strength meters," in *Proc. NDSS 2014*, pp. 1–16.

[10] S. Komanduri, R. Shay, P. G. Kelley *et al.*, "Of passwords and people: measuring the effect of password-composition policies," in *Proc. CHI 2011*, pp. 2595–2604.

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

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

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

[14] J. Blythe, R. Koppel, and S. W. Smith, "Circumvention of security: Good users do bad things," *IEEE Secur. & Priv.*, vol. 11, no. 5, pp. 80–83, 2013.

[15] D. Florêncio and C. Herley, "Where do security policies come from?" in *Proc. SOUPS 2010*, pp. 1–14.

[16] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from markov models," in *Proc. NDSS 2012*, pp. 1–15.

[17] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the j-pake password-authenticated key exchange protocol," in *Proc. IEEE S&P 2015*, pp. 571–587.

[18] L. Chen, H. W. Lim, and G. Yang, "Cross-domain password-based authenticated key exchange revisited," *ACM Trans. Inform. Syst. Secur.*, vol. 16, no. 4, pp. 1–37, 2014.

[19] D. Malone and K. Maher, "Investigating the distribution of password choices," in *Proc. WWW 2012*, pp. 301–310.

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

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

[22] E. H. Spafford, "Opus: Preventing weak password choices," *Computers & Security*, vol. 11, no. 3, pp. 273–278, 1992.

[23] D. Florêncio, C. Herley, and P. van Oorschot, "An administrators guide to internet password research," in *Proc. USENIX LISA 2014*, pp. 44–61.

[24] J. Blocki and A. Datta, "CASH: A cost asymmetric secure hash algorithm for optimal password protection," *CoRR*, vol. abs/1509.00239, 2015, http://arxiv.org/pdf/1509.00239v1.pdf.

[25] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password strength: an empirical analysis," in *Proc. INFOCOM 2010*, pp. 1–9.

[26] R. Bowes, *Password dictionaries*, Oct. 2011, https://wiki.skullsecurity.org/Passwords.

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

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

[29] S. Designer, *John the Ripper password cracker*, Feb. 1996, http://www.openwall.com/john/.

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

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

[32] B. Ur, S. M. Segreti, L. Bauer *et al.*, "Measuring real-world accuracies and biases in modeling password guessability," in *Proc. USENIX SEC 2015*, pp. 463–481.

[33] R. Martin, *Amid Widespread Data Breaches in China*, Dec. 2011, http://www.techinasia.com/alipay-hack/.

[34] L. Constantin, *Security Gurus 0wned by Black Hats*, July 2009, http://www.programdoc.com/1114_98794_1.htm.

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

[36] D. Goodin, *Personal data is exposed as a result of a five-month-old hack on 000Webhost*, Oct. 2015, http://t.cn/R4tKrEU.

[37] J. Mick, *Russian Hackers Compile List of 10M+ Stolen Gmail, Yandex, Mailru*, Sep. 2014, http://t.cn/R4tmJE3.

[38] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis on chinese web passwords," in *Proc. USENIX SEC 2014*, pp. 559–574.

[39] C. Johnston, *Why your password cant have symbols*, April 2013, http://arstechnica.com/security/2013/04/why-your-password-cant-have-symbols-or-be-longer-than-16-characters/.

[40] M. L. Pao, "An empirical examination of lotka's law," *J. Amer. Soc. Inform. Sci.*, vol. 37, no. 1, pp. 26–33, 1986.

[41] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[42] R. Veras, C. Collins, and J. Thorpe, "On the semantic patterns of passwords and their security impact," in *Proc. NDSS 2014*, pp. 1–16.

[43] M. Newman, "Power laws, pareto distributions and Zipf's law," *Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, 2005.

[44] T. Maillart, D. Sornette, S. Spaeth, and G. Von Krogh, "Empirical tests of zipf's law mechanism in open source linux distribution," *Phys. Rev. Lett.*, vol. 101, no. 21, pp. 701–714, 2008.

[45] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, 2000.

[46] R. M. Royall, "The effect of sample size on the meaning of significance tests," *The Amer. Statist.*, vol. 40, no. 4, pp. 313–315, 1986.

[47] Z. Huang, E. Ayday, J. Fellay, J.-P. Hubaux, and A. Juels, "Genoguard: Protecting genomic data against brute-force attacks," in *Proc. IEEE S&P 2015*, pp. 447–462.

Ding Wang received his B.S. Degree in Information Security from Nankai University in 2008. Currently, he is pursuing his Ph.D. degree at Peking University. He received two outstanding reviewer awards from Elsevier and authored an "ESI highly cited paper". His research interests mainly focus on password-based authentication and provable security.

Gaopeng Jian received his B.S. Degree in Fundamental Mathematics from Nankai university in 2011. Now he is pursuing his Ph.D. degree majoring in Cryptography at Peking university. His research interests include algebraic geometry codes, password authentication and lattice based cryptography.

Xinyi Huang received his Ph.D. degree from University of Wollongong, Australia. He is currently a Professor at Fujian Normal University, China, and the Co-Director of Fujian Provincial Key Laboratory of Network Security and Cryptology. His research interests include applied cryptography and network security. He is an associate editor of IEEE Transactions on Dependable and Secure Computing and has served as the program/general chair or TPC in over 80 international conferences.

Ping Wang received his B.S. degree from University of Electronic Science and Technology of China in 1983, and Ph.D. degree from University of Massachusetts, USA, in 1996. Now he is a Professor in Peking University. He served as TPC co-chairs of several security conferences. His research interests include system security and distributed computing.

APPENDIX A
HOW TO SET UP THE KOLMOGOROV-SMIRNOV TESTS FOR PASSWORD DATASETS

In most previous statistical studies regarding Kolmogorov-Smirnov (KS) test on power-law or Zipf's law distribution (e.g., [1]–[3]), the sample sizes are often relatively small (e.g., in hundreds or thousands), and the 0.05 significance level is generally preferred. This indicates that the 0.05 significance level can be used for the small password datasets (e.g., Myspace and Singles.org) in our study.

However, we also note that nine of our datasets consist of millions of passwords, and the 0.05 significance level is highly unlikely to be suitable for KS tests on these large datasets. This is due to the fact that "given a sufficiently large sample, extremely small and non-notable differences can be found to be statistically significant, and statistical significance says nothing about the practical significance of a difference" [4], which is known as the effect of sample size on the practical significance of a statistical test [5]. In addition, significance levels shall be set according to specific circumstances [4]. For instance, in genome association studies, significance levels are often as low as 10^{-8} for the million-sized genomes [6].

Accordingly, for the KS tests on large password datasets to be meaningful, the 0.05 significance level shall be adjusted to a much lower value. However, as far as we know, so far little attention has been paid to this issue. Now a natural way to overcome this issue is to reduce the sample size for each large dataset to a comparable one (i.e., $5 * 10^5$) with that of [1], [5], when performing the KS tests. More specifically, for each million-sized password dataset, we randomly draw $5*10^5$ passwords from it and then use the $5 * 10^5$ sampled passwords to perform a KS test against the fitted Zipf model, and obtain the corresponding p-value. To ensure validity, this process is repeated 1000 times for each dataset, and this would produce 1000 sampled p-values for each dataset. Finally, The p-value for each dataset (as shown in Table V of the main text) is an average of these 1000 sampled p-values.

APPENDIX B
IMPLICATIONS FOR PASSWORD CREATION POLICIES

Recently, many works on password policy (e.g., [7], [8]) 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 mere simple *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 4 of the main text, we have shown that in most cases, user-generated passwords well 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 [9], now $p(X = x)$ obeys a power law distribution. More specifically,

$$p(X = x) = C' \cdot x^{-\alpha}, \tag{1}$$

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

$$\begin{aligned} 1 &= \int_{x_{min}}^{\infty} p(X = x)\, dx \\ &= C' \cdot \int_{x_{min}}^{\infty} x^{-\alpha}\, dx \\ &= \frac{C'}{1-\alpha}[x^{-\alpha+1}]_{x_{min}}^{\infty}, \end{aligned} \tag{2}$$

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

$$C' = (\alpha - 1){x_{min}}^{\alpha-1}. \tag{3}$$

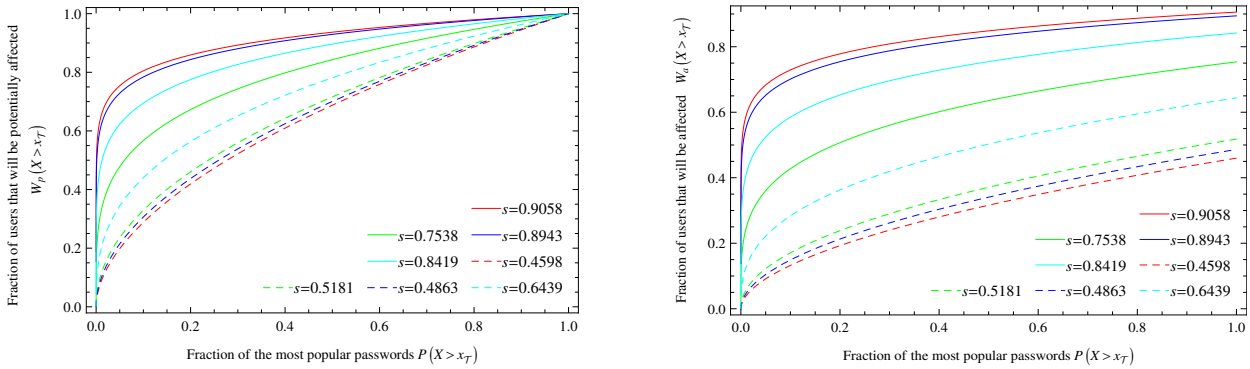
Thus, the probability that the frequency of a particular password will be greater than x ($x \geq x_{min}$) is given by

$$\begin{aligned} P(X > x) &= \int_{x}^{\infty} p(X = x')\, dx' \\ &= \frac{C'}{\alpha - 1} x^{-\alpha+1} = (\frac{x}{x_{min}})^{-\alpha+1}. \end{aligned} \tag{4}$$

Note that by definition, $P(X > x)$ can also be seen as the cumulative password popularity distribution function. Based on Eq. 2 and Eq. 3 as well as the fact that $\alpha = 1 + 1/s > 2$ (see s in Table V of the main text), the largest frequency $x_{\mathcal{T}}$ allowed under a threshold $\mathcal{T}$ can be determined

$$\begin{aligned} x_{\mathcal{T}} &= \mathcal{T} \cdot \int_{x_{min}}^{\infty} x p(X = x)\, dx \\ &= \mathcal{T} \cdot C' \cdot \int_{x_{min}}^{\infty} x^{-\alpha+1}\, dx \\ &= \mathcal{T} \cdot \frac{\alpha - 1}{\alpha - 2} x_{min}. \end{aligned} \tag{5}$$

We denote the exact fraction of user accounts (with password frequencies exceeding $x_{\mathcal{T}}$) that will be *potentially* and



(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. 1. 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 the main text.

TABLE I. 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.

actually affected by the threshold $\mathcal{T}$ to be $W_p(X > x_\mathcal{T})$ and $W_a(X > x_\mathcal{T})$,[1] respectively, where

$$W_p(X > x_\mathcal{T}) = \frac{\int_{x_\mathcal{T}}^{\infty} x' p(X = x')\, dx'}{\int_{x_{min}}^{\infty} x' p(X = x')\, dx'} = (\frac{x_\mathcal{T}}{x_{min}})^{-\alpha+2}. \tag{6}$$

$$\begin{aligned} W_a(X > x_\mathcal{T}) &= \frac{\int_{x_\mathcal{T}}^{\infty} (x' - x_\mathcal{T}) p(X = x')\, dx'}{\int_{x_{min}}^{\infty} x' p(X = x')\, dx'} \\ &= \frac{1}{\alpha - 1} \cdot (\frac{x_\mathcal{T}}{x_{min}})^{-\alpha+2}. \end{aligned} \tag{7}$$

Using Eqs. 4∼7, we can get the fraction of user accounts with each of its password lies in the most popular part $P(X > x_\mathcal{T})$:

$$W_p(X > x_\mathcal{T}) = (P(X > x_\mathcal{T}))^{(-\alpha+2)/(-\alpha+1)}. \tag{8}$$

Since $\alpha = 1 + 1/s$, Eq. 8 can be re-written as

$$W_p(X > x_\mathcal{T}) = (P(X > x_\mathcal{T}))^{(1-\frac{1}{s})/(-\frac{1}{s})} = (P(X > x_\mathcal{T}))^{1-s}. \tag{9}$$

Similarly, Eq. 7 can be re-written as

$$\begin{aligned} W_a(X > x_\mathcal{T}) &= \frac{1}{\alpha - 1} \cdot (P(X > x_\mathcal{T}))^{(1-\frac{1}{s})/(-\frac{1}{s})} \\ &= s \cdot (P(X > x_\mathcal{T}))^{1-s}. \end{aligned} \tag{10}$$

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. 1 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 of the main text.

The steep increase of W_p and W_a at the very beginning of their curves (see Fig. 1) explicitly reveal that, popular passwords are overly popular and a non-negligible fraction of users will be inconvenienced even if only a marginal proportion of popular passwords are checked. For example, according to Eq. 10, 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 I. 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

[1]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 $\mathcal{T} = 1/1024$.

large proportion of passwords that are not frequent (i.e., their frequencies are below x_{min}), which is generally called the "noisy tail" [2] 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.

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, previous works (e.g., [7], [10] just explicitly (or implicitly) suggested such a value for $\mathcal{T}$. On the other hand, the Zipf's law revealed in Section 4 of the main text 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 [11], 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.

Appendix C
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 [12] 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 [13]–[15] and also won the NSA 2013 annual "Best Scientific Cybersecurity Paper Award" [16]. Here we report an inherent flaw in its strength conversion and further manage to figure out how to fix it.

For better comprehension, here we follow the notations in [12] 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 $0<\alpha\leq 1$, $\mu_\alpha(\mathcal{X})$ = 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:

$$G_\alpha(\mathcal{X}) = (1 - \lambda_{\mu_\alpha}) \cdot \mu_\alpha + \sum_{i=1}^{\mu_\alpha} p_i \cdot i, \tag{11}$$

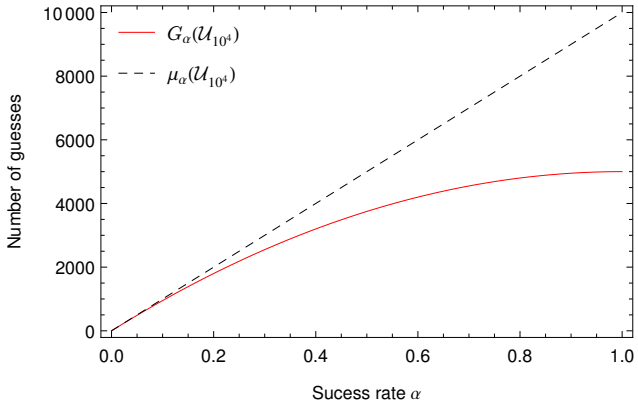


Fig. 2. How $G_\alpha(\mathcal{U}_N)$ and $\mu_\alpha(\mathcal{U}_N)$ vary with α. We use a discrete uniform distribution $\mathcal{U}_{10^4}$ (i.e., $p_i = 1/10^4$ for all $1 \leq i \leq 10^4$) as an example.

$G_\alpha(\mathcal{X})$ characterizes the expected number of guesses per account to reach a success rate α. The intuition of Eq. 11 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 [12] 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 [17] for a more detailed explanation):

$$\frac{G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} = \frac{\mathcal{N}+1}{2} \tag{12}$$

Then, it is natural to get $N = \left[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\right]$. According to the definition of the effective key-length: "$\cdots$ it represents the size of a randomly chosen cryptographic key which would give equivalent security." (see Section II-E of [12] and pp.49 of [17]), one can compute the effective key-length of $G_\alpha(\mathcal{X})$ as

$$\widetilde{G}_\alpha(\mathcal{X}) = \lg \mathcal{N} = \lg\left[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\right] \tag{13}$$

$\widetilde{G}_\alpha(\mathcal{X})$ obtained from Eq. 13 *should have been constant* for any uniform distribution $\mathcal{U}_\mathcal{N}$, but Bonneau [12] found it was not the case. So, he artificially added the "correction factor" $\lg \frac{1}{2-\lambda_{\mu_\alpha}}$ to $\widetilde{G}_\alpha(\mathcal{X})$, giving:

$$\widetilde{G}_\alpha(\mathcal{X}) = \lg\left[\frac{2 \cdot G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} - 1\right] + \lg \frac{1}{2 - \lambda_{\mu_\alpha}} \tag{14}$$

However, in the following we will demonstrate that Eq. 12 inherently does not hold true. As a result, Eq. 13 is invalid. As can be seen from Fig.2(a) in [12], it was believed that $G_\alpha(\mathcal{U}_\mathcal{N}) = \mu_\alpha(\mathcal{U}_\mathcal{N})$. Quite the contrary, our Fig. 2 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. 11, one can get

$$\begin{aligned} G_\alpha(\mathcal{U}_\mathcal{N}) &= \sum_{i=1}^{\mu_\alpha} i \cdot \frac{1}{\mathcal{N}} + (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha \\ &= \frac{(1+\mu_\alpha)\mu_\alpha}{2} \cdot \frac{1}{\mathcal{N}} + (1-\lambda_{\mu_\alpha}) \cdot \mu_\alpha \end{aligned} \tag{15}$$

On the other hand, according to the definition of μ_α and λ_β in [12], we get

$$\mu_\alpha(\mathcal{U}_\mathcal{N}) = \mathcal{N} \cdot \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}) \tag{16}$$

Based on Eq. 16, Eq. 15 can be rewritten as

$$\begin{aligned} G_\alpha(\mathcal{U}_\mathcal{N}) &= \frac{(1+\mathcal{N}\cdot\lambda_{\mu_\alpha})\cdot\mathcal{N}\lambda_{\mu_\alpha}}{2\mathcal{N}} + (1-\lambda_{\mu_\alpha})\cdot\mathcal{N}\cdot\lambda_{\mu_\alpha} \\ &= \frac{\lambda_{\mu_\alpha}}{2} + \frac{1}{2}(2-\lambda_{\mu_\alpha})\cdot\mathcal{N}\cdot\lambda_{\mu_\alpha} \end{aligned} \tag{17}$$

From Eq. 16 and Eq. 17, it is evident that $G_\alpha(\mathcal{U}_\mathcal{N}) \neq \mu_\alpha(\mathcal{U}_\mathcal{N})$. Based on Eq. 17, for $\mathcal{U}_\mathcal{N}$ and $\mathcal{X}$ to be of equivalent security, we get

$$G_\alpha(\mathcal{U}_\mathcal{N}) = G_\alpha(\mathcal{X}) \overset{\text{Eq.16}}{\Longrightarrow} \mathcal{N} = \frac{2G_\alpha(\mathcal{X}) - \lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})}{(2-\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N}))\lambda_{\mu_\alpha}(\mathcal{U}_\mathcal{N})} \tag{18}$$

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 [12]), $\widetilde{G}_\alpha(\mathcal{X})$ 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 14 datasets. Further, for human-generated passwords, generally $\mathcal{N} \geq 2^{15}$ [12], [18]. 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 [12] leave out the distribution $\mathcal{X}$ or $\mathcal{U}_\mathcal{N}$. Based on this observation and Eq. 18, for $0 < \alpha < 1$, the Eq. 12 can be shown to be incorrect:

$$\frac{G_\alpha(\mathcal{X})}{\lambda_{\mu_\alpha}} = \frac{1+\mathcal{N}\cdot(2-\lambda_{\mu_\alpha})}{2} \neq \frac{\mathcal{N}+1}{2} \tag{19}$$

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

Further, using Eq. 18, the "effective key-length" (i.e., bit-strength) of $G_\alpha(\mathcal{X})$ can be naturally formulated as

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

It follows that there is no need to add a factitious "correction factor" in the strength conversion of α-guesswork (i.e., when converting $G_\alpha(\mathcal{X})$ to its effective key-length form $\widetilde{G}_\alpha(\mathcal{X})$), thereby demonstrating the inherent flaw in [12], [17]. As far as we know, little public work has ever employed the metric $G_\alpha(\mathcal{X})$, and most related works (e.g., [13]–[15], [18]–[20]) have preferred the effective key-length metric $\widetilde{G}_\alpha(\mathcal{X})$. While $\widetilde{G}_\alpha(\mathcal{X})$ is overwhelmingly favored over $G_\alpha(\mathcal{X})$ in the research community 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 for Appendix

[1] A. Clauset, C. R. Shalizi, and M. E. Newman, "Power-law distributions in empirical data," *SIAM review*, vol. 51, no. 4, pp. 661–703, 2009.

[2] M. Newman, "Power laws, pareto distributions and zipf's law," *Contemporary Physics*, vol. 46, no. 5, pp. 323–351, 2005.

[3] M. L. Pao, "An empirical examination of lotka's law," *J. Amer. Soc. Inform. Sci.*, vol. 37, no. 1, pp. 26–33, 1986.

[4] S. Nolan, *Study Guide for Essentials of Statistics for the Behavioral Sciences*. Worth Publishers, 2013.

[5] R. M. Royall, "The effect of sample size on the meaning of significance tests," *The Amer. Statist.*, vol. 40, no. 4, pp. 313–315, 1986.

[6] P. Sham and S. Purcell, "Statistical power and significance testing in large-scale genetic studies," *Nature Genetics*, vol. 15, no. 5, pp. 335–346, 2014.

[7] S. Schechter, C. Herley, and M. Mitzenmacher, "Popularity is everything: A new approach to protecting passwords from statistical-guessing attacks," in *Proc. HotSec 2010*, pp. 1–8.

[8] C. Castelluccia, M. Dürmuth, and D. Perito, "Adaptive password-strength meters from markov models," in *Proc. NDSS 2012*, pp. 1–15.

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

[10] D. Florêncio, C. Herley, and P. van Oorschot, "An administrators guide to internet password research," in *Proc. USENIX LISA 2014*, pp. 35–52.

[11] W. Burr, D. Dodson, R. Perlner, W. Polk, S. Gupta, and E. Nabbus, "NIST SP800-63-2 – electronic authentication guideline," NIST, Reston, VA, Tech. Rep., Aug. 2013.

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

[13] R. Chatterjee, J. Bonneau, A. Juels, and T. Ristenpart, "Cracking-resistant password vaults using natural language encoders," in *Proc. IEEE S&P 2015*, pp. 481 – 498.

[14] J. H. Huh, S. Oh, H. Kim, K. Beznosov, A. Mohan, and S. R. Rajagopalan, "Surpass: System-initiated user-replaceable passwords," in *Proc. CCS 2015*, pp. 170–181.

[15] B. B. Zhu, J. Yan, D. Wei, and M. Yang, "Security analyses of click-based graphical passwords via image point memorability," in *Proc. CCS 2014*, pp. 1217–1231.

[16] *1st Annual Best Scientific Cybersecurity Paper Competition*, July 2013, http://cps-vo.org/group/sos/papercompetition2012.

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

[18] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis on chinese web passwords," in *Proc. USENIX Security 2014*, Aug., pp. 559–574.

[19] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz, "Quantifying the security of graphical passwords: The case of android unlock patterns," in *Proc. CCS 2013*, pp. 161–172.

[20] Y. Song, G. Cho, S. Oh, H. Kim, and J. Huh, "On the effectiveness of pattern lock strength meters: Measuring the strength of real world pattern locks," in *Proc. CHI 2015*, pp. 2343–2352.