Zipf's Law in Passwords

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

***Abstract*— Despite three decades of intensive research efforts, it remains an open question as to what is the underlying distribution of user-generated passwords. In this paper, we make a substantial step forward toward understanding this foundational question. By introducing a number of computational statistical techniques and based on 14 large-scale data sets, which consist of 113.3 million real-world passwords, we, for the first time, propose two Zipf-like models (i.e., PDF-Zipf and CDF-Zipf) to characterize the distribution of passwords. More specifically, our PDF-Zipf model can well fit the popular passwords and obtain a coefficient of determination larger than 0.97; our CDF-Zipf model can well fit the entire password data set, with the *maximum* cumulative distribution function (CDF) deviation between the empirical distribution and the fitted theoretical model being 0.49%∼4.59% (on an average 1.85%). With the concrete knowledge of password distributions, we suggest a new metric for measuring the strength of password data sets. Extensive experimental results show the effectiveness and general applicability of the proposed Zipf-like models and security metric.**

***Index Terms*— User authentication, password distribution, Zipf's law, strength metric, password policy.**

I. Introduction

PASSWORD-BASED authentication continues to be the dominant mechanism of user authentication over the Internet. Despite its ubiquity, this kind of authentication is accompanied by the dilemma of generating passwords that are challenging for powerful attackers to crack but easy for common users to remember. Truly random passwords are difficult for users to memorize, while user-generated 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]). This means these passwords are drawn from a rather small space, and thus they are prone to guessing attacks.

To mitigate this notorious "security-usability" dilemma, various password creation policies have been proposed, e.g., random generation [4], rule-based [5], entropy-based [6] and cracking-based [7]. They force newly created passwords to adhere to some rules and to achieve an acceptable strength. The diversity of password strength meters and rules brings about an enormous variety of requirements between different web services, resulting in highly conflicting strength outcomes for the same password [8]. 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 [8], [9]) are direct results of the inconsistent password strength meters employed by different web services. They may be further explained by the un-soundness of current password meters and the diverse interests of each website. It is a rare piece of good news in password research that password policies do impact user password choices and, if well-designed, password policies can significantly improve password security while maintain usability [10]. Accordingly, much attention (e.g., [8], [11], [12]) has been paid to the design and analysis of password policies. While stricter policies might make passwords harder to crack, the side effect is that users may feel harder to create and 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 they will impact negatively on user productivity. Ultimately, users will try every means to circumvent such un-friendly policies.

As a result, different types of application systems typically have quite different choices. For e-commerce sites like eBay, 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 less restrictive password policies [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 digits and symbols, and rejection of popular passwords like `pa$$word123`).

As different systems often implement quite varied password policies, a number of critical issues arise: how can the password policy designers evaluate their policies? how can

Manuscript received February 6, 2017; revised June 15, 2017; accepted June 21, 2017. Date of publication June 28, 2017; date of current version August 22, 2017. This work was supported in part by the National Key Research and Development Plan under Grant 2016YFB0800603 and Grant 2017YFB1200704 and in part by the National Natural Science Foundation of China under Grant 61472016 and Grant 61472083. This paper was presented in part at the Proceeding of the 21th European Symposium on Research in Computer Security (ESORICS 2016). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Karen Renaud. *(Corresponding author: Ping Wang.)*

D. Wang, H. Cheng, and G. Jian are with the School of EECS, Peking University, Beijing 100871, China, and also with the Key Laboratory of High Confidence Software Technologies, Ministry of Education, Beijing 100871, China (e-mail: wangdingg@pku.edu.cn; chenghaibo@pku.edu.cn; gpjian@pku.edu.cn).

P. Wang is with the National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China, and also with the School of Software and Microelectronics, Peking University, Beijing 100871, China (e-mail: pwang@pku.edu.cn).

X. Huang is with the School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China (e-mail: xyhuang81@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2017.2721359

the administrators select the right policy for their systems? In addition, usually the users of a system (as well as its services) may dynamically change as time goes on. This highly leads to large variations in the password dataset (distribution) after some period of time (e.g., one year) even though the password policy stays the same, which is especially true for Internet-scale service providers. In this situation, it is desirable that security administrators quantify the strength of passwords and, accordingly, adjust the password policy. Either failing to notice the changes in the password distribution, 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 distribution 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 crux of designing, selecting and adjusting an appropriate password policy lies in how to accurately assess the strength of a password distribution created under this policy. Note that, here we presume that each existing authentication system has already adopted some password policy (e.g., [7], [16]), and its adjustment mainly involves changing some rules and the password strength threshold.

Inevitably, the accomplishment of accurately assessing the strength of a password distribution would entail the settlement of a more fundamental question: how to precisely characterize a given password distribution? Or equally, *what is the distribution that user-generated passwords follow*? Despite more than 30 years of intensive research efforts, this same old question is asked year in year out. This may well explain why most of today's password-based cryptographic protocols (in hundreds, some recent ones include [17], [18]) still rely on a far from realistic 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] Our PDF-Zipf model is based on the efforts of Malone-Maher's approach [19], but with the difference that we do not fit these unpopular passwords of a dataset into the model. We further propose another model called CDF-Zipf. Both our models reach a different conclusion with that of [19].

A. Our Contributions

In this work, we make the following key contributions:

1) Two Zipf Models: We propose two Zipf-like models to characterize the distribution of passwords. Our results consistently show that Zipf's law exists in real-life passwords. More specially, Our PDF-Zipf model works on popular passwords and reveals that: (1) the vulnerable portion of user-chosen passwords (i.e., popular passwords with a frequency $f \geq 4$) *naturally* follow a Zipf-distribution; and (2) the remaining portion of user-chosen passwords (i.e., unpopular passwords with a frequency $f \leq 3$) are *highly likely* to follow a Zipf-distribution. Based on the PDF-Zipf model, we further develop an advanced model, called CDF-Zipf, that works on the entire password dataset and yields much better fitting results: the maximum distance between the cumulative distribution function (CDF) of the real data and CDF of the fitted model is 0.48%∼4.59% (avg. 1.84%), while this figure for the PDF-Zipf model is 6.26%∼27.88% (avg. 16.56%).

2) A Security Metric: We propose a novel metric for measuring the security strength of a given password distribution. This metric utilizes the *concrete* knowledge of the password distribution function, and thus it overcomes various problems in existing security metrics (e.g., uncertainties in cracking-based approaches [16] and non-deterministic nature in α-guesswork [20]). Our metric facilitates a better grasp of the strength of a given password dataset (either in plain-text or hashed form) in a mathematically rigorous manner. This enables security administrators to better evaluate the security property of a password policy under which the password dataset is created.

3) An Extensive Evaluation: Our evaluation builds on 14 large-scale real-world password datasets. Our datasets are composed of a total of 113.3 million passwords, cover a diversity of Internet services, involve various languages/password creation policies, and are among the largest corpuses ever collected for a password study. Results from extensive experiments suggest 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] and [20] that user passwords are "unlikely to actually be Zipf distributed".

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, aiming 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

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

to shape password creation, among which is the influential NIST Electronic Authentication Guideline [6]. 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. Later, Houshmand and Aggarwal [16] proposed a novel policy that improves password security while maintaining usability: it first analyzes whether a user-chosen password is weak or strong according to the empirical cracking-based results, and then slightly modifies and strengthens the password if it is weak. This policy facilitates measuring the strength of individual passwords more accurately. In addition, it can be adjusted more flexibly than previous policies, because its adjustment only involves tuning the threshold within a continuous range. Observing that users often reuse or slightly modify an existing password for a new service but not create a completely new password, Wang *et al.* [7] further improves Houshmand-Aggarwal's policy to more practically capture user behavior.

Perhaps the most relevant policy related to our strength metric for assessing password datasets (see Section V) is suggested by Schechter *et al.* [5]. 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 the server (or a dishonest insider of the server) from learning the queried password 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 [5]), yet how it would affect usability has not been evaluated thoroughly. No theoretical or empirical usability results have ever been reported.

Based on the proposed Zipf's law, we manage to develop theoretical models to predict that, as an immediate consequence of Schechter *et al.* policy [5], a large fraction of users might be annoyed by forbidding them to use their intended passwords that are typically popular. For instance, 34.89% of users in www.tianya.cn use passwords that are more frequent than $\mathcal{T} = \frac{1}{10^6}$, indicating 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 to some extent (but not readily [3]) counteracted by the server by using non-cryptographic techniques, such as modern machine-learning-based detecting, rate-limiting or locking strategies [23]. In contrast, offline guessing is performed on local machine that is fully under the attacker's control, and thus she can make as many guesses as possible given enough time and computational power.

Florencio *et al.* [24] discussed scenarios where offline guessing constitutes a real threat, and they 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. It is neither related to the particular nature of the system (e.g., which hash function is used, SHA-1, PBKDF2 or CASH [25]?) 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 [26]. 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 remembrable [1], [27]. However, users rarely use unmodified elements from such lists, for instance, because password policies prevent this practice. 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 [28]. This sort of techniques has emerged as early as 1979 in Morris-Thompson's analysis of 3,000 passwords [29]. This initial work has been followed by independent works [21], [30]. Later on, some dedicated software tools like John the Ripper (JTR) [31] appeared. Subsequent studies (e.g., [10], [11]) have often employed these automated software tools to conduct dictionary attacks as a secondary goal.

It was not until very recently that password cracking began to evolve from art to science. Narayanan and Shmatikov [32]

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
000webhost	Web hosting	USA	English	**Oct. 28, 2015**	PHP programming bug	15,251,073	10,583,709
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
Hak5	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
Battlefield	Gaming	USA	English	June 26, 2011	Hacker breached	542,386	417,453
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
Flirtlife.de	Dating	Germany	German	**May 25, 2006**	Hacker breached	343,064	115,589

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.* [30] suggested a novel technique for automatically deriving word-mangling rules. 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. It is able to crack 28% to 129% more passwords than JTR [31].

In 2014, Ma *et al.* [33] introduced natural language processing techniques, such as smoothing and normalization into Markov-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-based cracking algorithms would perform better than PCFG-based cracking algorithms.

In 2015, Ur *et al.* [34] investigated: (1) how the above cracking algorithms used by researchers compare to real-world cracking by professionals; and (2) how the choice of cracking algorithms influences research conclusions. They found that each cracking algorithm is highly sensitive to its configuration. They also observe 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 (distribution)* can rely on a single algorithm.

In 2016, Wang *et al.* [3] systematically investigated to what extent an online guessing attacker can gain advantages by making use of various types of user personal information, such as leaked passwords and user demographic information. They devised a general guessing framework, called TarGuess, that incorporates seven sound probabilistic guessing models TarGuess-I∼VII. Their work provides comprehensive, quantitative evidence of how serious the threat of targeted online password guessing is. For instance, TarGuess-III can gain success rates as high as 73% with just 100 guesses against normal users and 32% against security-savvy hackers.

III. DATASETS, LINEAR REGRESSION AND KS TEST

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], [33], [34]). 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 is 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

TABLE II
CHARACTER COMPOSITION INFORMATION ABOUT EACH PASSWORD DATASET

Dataset	[a-z]+	[A-Z]+	[A-Za-z]+	[0-9]+	[a-zA-Z0-9]+	[a-z]+[0-9]+	**[a-z]+1**	[a-zA-Z]+[0-9]+	[0-9]+[a-zA-Z]+	[0-9]+[a-z]+
Tianya	9.96%	0.18%	10.29%	**63.77%**	98.05%	14.63%	0.12%	15.64%	4.37%	4.11%
Dodonew	8.79%	0.27%	9.37%	**20.49%**	82.88%	**40.81%**	1.39%	42.94%	7.31%	6.95%
CSDN	11.64%	0.47%	12.35%	**45.01%**	96.31%	26.14%	0.24%	28.45%	6.46%	5.88%
000webhost	**0.04%**	**0.00%**	0.26%	**0.02%**	**93.08%**	54.42%	4.66%	60.95%	8.43%	7.28%
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 III
LENGTH DISTRIBUTION INFORMATION OF EACH DATASET

Length	1-3	4	5	6	7	8	9	10	11	12	13-16	17-30	30+	All
Tianya	0.61%	0.65%	0.55%	33.77%	13.92%	18.10%	9.59%	10.28%	5.53%	2.88%	4.05%	0.07%	0.00%	100%
Dodonew	0.36%	0.70%	0.78%	9.71%	13.45%	18.49%	20.29%	14.69%	3.10%	1.34%	10.24%	6.79%	0.04%	100%
CSDN	**0.01%**	**0.10%**	**0.51%**	**1.29%**	**0.26%**	36.38%	24.15%	14.48%	9.78%	5.75%	6.96%	0.32%	0.00%	100%
000webhost	**0.00%**	**0.00%**	**0.01%**	5.70%	7.92%	5.70%	7.92%	21.81%	15.41%	14.51%	10.49%	7.67%	14.35%	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	0.55%	1.77%	2.49%	**16.00%**	**10.50%**	**16.00%**	**10.50%**	**23.95%**	**14.60%**	**14.28%**	5.92%	3.64%	5.49%	100%

China in December, 2011 [35] 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.

The fourth dataset 000webhost contains 15.3 million passwords that were leaked in plain-text from 000Webhost, a popular free web-hosting site. This breach has been confirmed by 000webhost officials, and it is believed to be the result of hackers who exploited a weakness in an old version of the PHP programming language. The fifth dataset is the 41.5K "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 [28] 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 a 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) [36]. 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, Battlefield, Gmail, Yandex.ru, Mail.ru and Flirtlife.de) to establish 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 32M passwords leaked from the gaming forum Rockyou in Dec. 2009 [37]; 542K Battlefield passwords were leaked by the hacker group LulzSec in 2011 [38]; The next three lists (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 [39]. It is said that these credentials are collected not by hacking the three sites but through phishing and other forms of attacks on users (e.g., keyloggers). The last dataset was leaked from the German dating site Flirtlife.de in May 2006, and about half of the accounts are still alive when the leakage was first detected [40].

B. Statistics About User-Chosen Passwords

The character composition information is summarized in Table II. Chinese users are more likely to use only digits to construct their passwords, while English users prefer using letters. This complies with [41]. 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 III shows the length distributions of each dataset. We can see that the most popular password lengths are between 6 and 10, which on average 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 exists in many financial companies [42], 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. 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 [43], [44]. 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. $D \in [0, 1]$ is essentially the max gap between the two CDF curves $F_n(x)$ and $F(x)$, the smaller the better.

This statistic D can be adopted to conduct a rigorous test. Note that, in our work the underlying distribution of each password dataset is itself determined by fitting the data and hence varies from one dataset to another. In other words, the password distribution is not fixed. Thus, we *cannot* compute the p-value by $\Pr(\sqrt{n}D > x) = 2\sum_{i=1}^{\infty}(-1)^{i-1}e^{-2i^2x^2}$. Instead, we need to resort to the Monte Carlo approach as recommended in [44]: (1) fit a given password dataset by using the theoretic model under question (e.g., PDF-Zipf), and compute the corresponding KS statistic D; (2) generate a number of (e.g., 2500 as suggested) synthetic datasets by using the same distribution parameters fitted from the empirical dataset; (3) fit each synthetic dataset individually by using this theoretic model, and calculate the corresponding D'; (4) p-value is defined to be fraction of the synthetic distances (i.e., D's) that are larger than D.

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

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

We now propose two theoretic Zipf-like models to characterize the distribution of passwords.

A. Our PDF-Zipf Model

Initially, probabilistic context-free grammar (PCFG) is a machine learning technique used in natural language processing (NLP), yet Weir *et al.* [30] 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 [45] and in dealing with the sparsity problem in passwords [33].

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 [44], software packages [47] and the Internet topology [48].

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:

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

[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 [46].

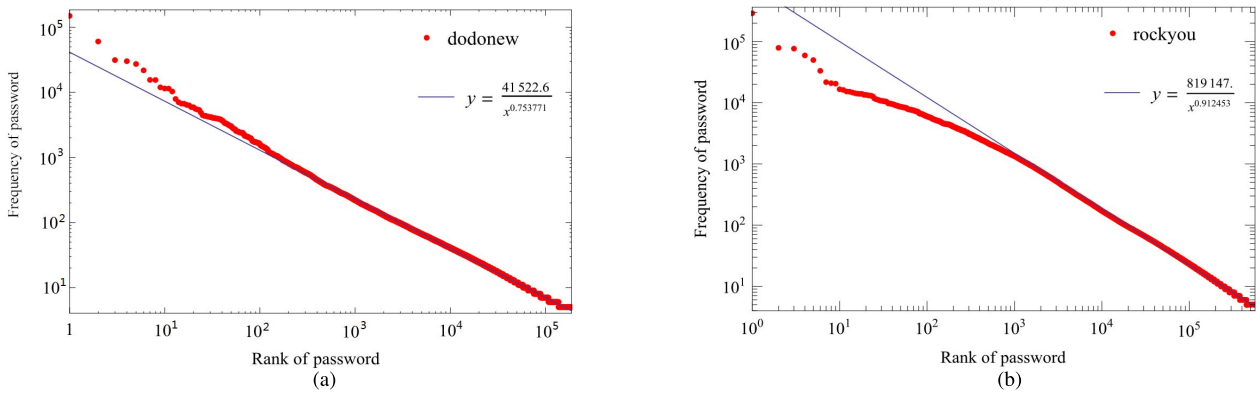


Fig. 1. Fitting passwords with our PDF-Zipf model. Dodonew includes passwords of Chinese users, while Rockyou includes passwords of English users. Though a few top-popular passwords do not lie on the fitting line, they are negligible as compared to the ones that dwell on the fitted line. (a) 16M Dodonew passwords: $R^2 = 0.996$. (b) 32M Rockyou passwords: $R^2 = 0.997$.

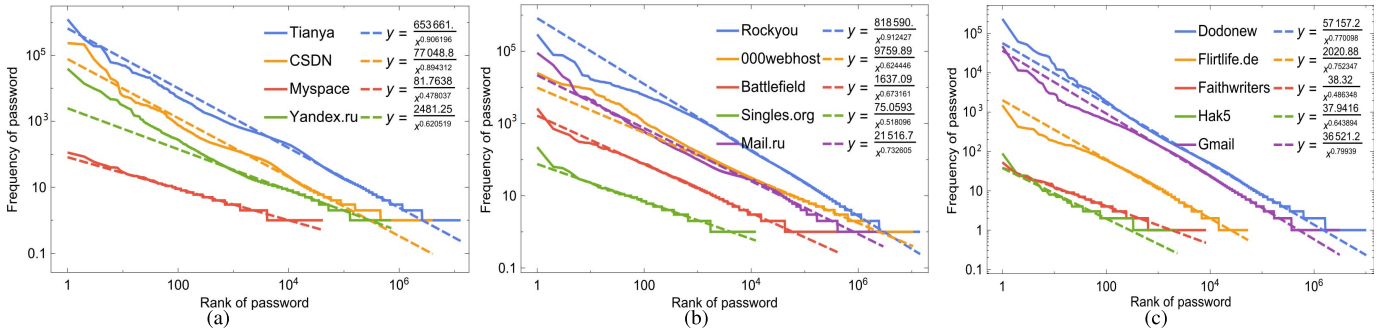


Fig. 2. Fitting passwords with our PDF-Zipf model, plotted on a log-log scale. For detailed Zipf parameters, see Table IV. To minimize the overlap among the fitting curves, we divide 14 datasets into the current three groups. (a) Zipf's law in passwords of Tianya, CSDN, Myspace and Yandex.ru. (b) Zipf's law in passwords of Rockyou, 000webhost, Battlefield, Singles.org and Mail.ru. (c) Zipf's law in passwords of Dodonew, Flirtlife.de, Faithwriters, Hak5 and Gmail.

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 line explains 99.55% of the fitted data points. 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.

More precisely, as summarized by the "Coefficient of determination" column in Table IV, every regression (except for Hak5) 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 IV, 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.

However, we can not conclude that the distribution of passwords is exactly characterized by the fitted theoretical Zipf models. Indeed, the KS tests are all with a sufficiently *low* p-value, and we have enough confidence to reject the hypothesis that our datasets are drawn from the modeled distribution (see parameters in Table IV). Note that, this does not contradict with our conclusion that user passwords follow the Zipf's law: if the Zipf parameters are computed more accurately, it is possible that we can obtain sufficiently *large* p-values that pass the KS tests. In Sec. IV-C, we show this by providing an improved Zipf model over the PDF-Zipf model.

TABLE IV
OUR PDF-ZIPF MODEL: FITTING RESULTS OF FOURTEEN PASSWORD DATASETS USING LINEAR REGRESSION ("PWs" STANDS FOR PASSWORDS)

Dataset	Total 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	0.161718	$< 10^{-4}$
Dodonew	16,231,271	5	0.21640911	187,901	0.753771	$4.618284 - 0.753771 * \log x$	0.995530686	0.164640	$< 10^{-4}$
CSDN	6,428,287	5	0.29841262	57,715	0.894307	$4.886747 - 0.894307 * \log x$	0.985106832	0.268982	$< 10^{-4}$
000webhost	15,251,073	5	0.19687867	229,725	0.624446	$7.354124 - 0.624446 * \log x$	0.989437653	0.111546	$< 10^{-4}$
Myspace	41,545	3	0.08094836	706	0.459808	$1.722674 - 0.459808 * \log x$	0.965861431	0.126651	$< 10^{-4}$
Singles.org	16,250	3	0.22135384	658	0.518096	$1.875405 - 0.518096 * \log x$	0.970277755	0.100741	$< 10^{-4}$
Faithwriters	9,709	3	0.12472963	242	0.486348	$1.583425 - 0.486348 * \log x$	0.974175889	0.140562	$< 10^{-4}$
Hak5	2,987	3	0.15400067	76	0.643896	$1.579116 - 0.643896 * \log x$	0.922662999	0.278761	$< 10^{-4}$
Rockyou	32,603,388	5	0.49600581	563,074	0.912453	$5.913362 - 0.912453 * \log x$	0.997298647	0.193567	$< 10^{-4}$
Battlefield	542,386	3	0.20773582	15,178	0.673161	$3.214071 - 0.673161 * \log x$	0.98384909	0.209718	$< 10^{-4}$
Gmail	4,926,650	5	0.29617143	77,397	0.799443	$4.903847 - 0.799443 * \log x$	0.995817202	0.217463	$< 10^{-4}$
Mail.ru	4,938,663	5	0.33034872	83,914	0.732600	$4.332851 - 0.732599 * \log x$	0.970047769	0.168754	$< 10^{-4}$
Yandex.ru	1,261,810	5	0.34210777	26,003	0.620519	$3.394671 - 0.620519 * \log x$	0.972507203	0.097279	$< 10^{-4}$

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 [44, Fig. 3]): 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., $x_{max}/x_{min} \geq 10^{2\sim3}$) [47]. Except for Hak5, this condition is satisfied by all our regressions.

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 [44]) that are with $s > 1$.

B. Justification for Our PDF-Zipf Methodology

In the above, we have shown that the distribution of popular passwords(e.g., with $f \geq 5$) can be approximated by the Zipf's law. In the following, we justify our methodology and provide evidence to support the conjecture that this law is highly likely to hold in the remaining part of user-generated passwords.

Malone and Maher [19] have also attempted to investigate password distributions. They concluded that their datasets (including 32M Rockyou) are "unlikely to actually be Zipf distributed". They also reported that "while a Zipf distribution does not fully describe our data, it provides a reasonable model, particularly of *the long tail* of password choices." Our PDF-Zipf model is based the efforts of Malone and Maher [19], but it differs from Malone-Maher's approach [19] in that these unpopular passwords (e.g., with $f_r < 3$) of a dataset are not fitted to the Zipf model, and we observe different results. In the following, we make an attempt to figure out why this work and Malone-Maher's work have different observations.

Unpopular passwords (e.g., with $f_r < 3$) constitute a non-negligible fraction of each dataset (see Table IV) and become the long tail of password choices (see [19, Fig. 1]) or the "noisy tail" in the statistical domain [46], 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 $\sigma = \sqrt{p_{pw_i}(1 - p_{pw_i})}$ [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 $\mu = p_{pw_i} \cdot |\mathcal{DS}|$ and $\sigma = \sqrt{p_{pw_i}(1 - p_{pw_i}) \cdot |\mathcal{DS}|}$, where f_{pw_i} is the frequency of pw_i in the password dataset $\mathcal{DS}$. Because generally $1 - p_{pw_i} \approx 1$, 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 different 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 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 $|X| = 62^{16} - 62^8 \approx 4.8*10^{28}$ 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 into the PDF-Zipf model, 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) exhibit a good Zipf property. This reveals one of the inherent limitations of the PDF-Zipf model: it is only suitable for characterize the popular passwords of a dataset. In Section IV-D, we further show that the PDF-Zipf model is also not accurate. Both limitations call for a better model.

It should be noted that, though these least frequent passwords do not *naturally* show the Zipf behavior, this fact does not contradict our conjecture that *the password population (of a site) is highly likely to follow a Zipf distribution*. From Table IV one can see that, generally, the larger the dataset is (see the second column), the larger the fraction of popular passwords (i.e., passwords to be used in regression, see the fourth column) will be. Based on this trend, one can expect that, had the dataset been sufficiently large, popular passwords would account for an overwhelming fraction (and unpopular passwords will be a small portion), and thus whether excluding these unpopular passwords or not would have little impact on the goodness of the fitting. That is, the entire dataset will exhibit a Zipf property.

To further justify our conjecture that user-generated password datasets[3] follow the 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 $120(= 3 \cdot 5 \cdot 8)$ 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 V 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 V are with decimals, because they are averaged over 1000 repeated experiments.

[3]Note that, a password dataset is a multi-set, and it can be seen as password samples randomly drawn from the underlying password distribution of a given authentication system.

TABLE V

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 more details and the 120 complete experiments, readers can see http://t.cn/R4ccgiF.

Our results on 120 experiments show that, given a perfect 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 and they could impair the overall fitting even if

popular events actually follow the Zipf's law. 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 conjecture that the distribution of the entire password dataset can be well approximated by the Zipf's law.

C. Our CDF-Zipf Model

As shown in Table IV and Fig. 2, an undesirable feature of the PDF-Zipf model is that, it can not well capture the distribution of the least popular passwords. Note that, due to the law of large numbers, the least frequent passwords are inherently difficult to be captured by a theoretic model that directly employs the probability of a password. We have tried various means to adjust the PDF-Zipf parameters fitted from the popular passwords to approximate these least popular passwords, yet we are always caught in a dilemma: if they are well captured, the overall fitting gets bad; if they are not considered, the overall fitting will be with large R^2s. Still, *the PDF-Zipf model (as well as [19]) provides us with a glimmer that passwords are likely to follow a Zipf-like distribution, and the key issue left is how to propose new Zipf models that overcome the limitations of the PDF-Zipf model.*

As the PDF of a distribution and its CDF can be converted to each other, why not directly model the CDF of a password distribution? Interestingly, we find the CDF graph of each *entire* dataset can be well fitted by the Zipf's law (see the dash-dot green lines in Figs. 3(a) to 3(c) and Figs. 3(g) to 3(i)). We call this model the CDF-Zipf model:

$$F_r = C' \cdot r^{s'}, \tag{3}$$

where F_r is the cumulative frequency of passwords up to rank r, C' and s' are constants depending on the password dataset and can be calculated by linear regression. $F_r(\cdot)$ is a step function, because $r = 1, 2, 3, \cdots$. Thus, we have

$$f_r = F_r - F_{r-1} = C' \cdot r^{s'} - C' \cdot (r-1)^{s'}. \tag{4}$$

Note that, f_r can be approximated by using the derivative of F_r when seeing F_r as a continuous function: $f_r \approx d(F_r)/dr = C' \cdot s' \cdot r^{s'-1}$, implying a Zipf's law.

We fit the CDF-Zipf model to our 14 datasets (see Fig. 4), and always obtain better fittings than the PDF-Zipf model in terms of the KS statistic D (i.e., the max gap between the CDF curves of a fitted model and the real data). Our CDF-Zipf parameters are calculated by linear regression using the well-known golden-section-search method.

As summarized in Table VI, the KS statistic D from fittings under our CDF-Zipf model is 0.006170∼0.045874 (avg 0.018457). It is invariably *smaller* than the corresponding D of the PDF-Zipf model (see Table IV). This means that the max CDF gap under the CDF-Zipf model is always smaller than those of the PDF-Zipf model. KS p-value results show that, for 9 datasets, we have enough confidence to reject the hypothesis that they are drawn from the exactly modeled distribution (see parameters in Table VI). As said earlier, this does *not* contradict with our conclusion that password distributions can be well approximated by Zipf's law. Actually, the low p-values are 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" [49], which is known as the effect of sample size on the practical significance of a statistical test [50].

TABLE VI
OUR CDF-ZIPF MODEL: FITTING RESULTS OF FOURTEEN DATASETS

	C'	s'	Dataset coverage	Statistic D	p-value
Tianya	0.062239	0.155478	100.00%	0.022798	0.0052
Dodonew	0.019429	0.211921	100.00%	0.004926	0.0124
CSDN	0.058799	0.148573	100.00%	0.022319	$<10^{-4}$
Rockyou	0.037433	0.187227	100.00%	0.045874	$<10^{-4}$
000webhost	0.005858	0.281557	100.00%	0.006170	$<10^{-4}$
Battlefield	0.010298	0.294932	100.00%	0.010557	0.0032
Flirtlife.de	0.034577	0.291596	100.00%	0.036448	$<10^{-4}$
Myspace	0.005646	0.403400	100.00%	0.008262	0.4440
Singles.org	0.020122	0.337620	100.00%	0.013786	0.1048
Faithwriters	0.013588	0.364763	100.00%	0.014524	0.2076
Hak5	0.029029	0.351792	100.00%	0.018078	0.5104
Mail.ru	0.025211	0.218212	100.00%	0.020773	0.0016
Gmail	0.020963	0.225653	100.00%	0.020543	0.0048
Yandex.ru	0.044248	0.197234	100.00%	0.013345	$<10^{-4}$

We have tested that, when confining each password sample to a comparable one (e.g., 10^4 or 10^5) with that of [44] and [50], most of the CDF-Zipf based fittings will pass the KS tests. This is corroborated by Table VI that four fittings on small datasets (i.e., Myspace, Singles.org, Faithwriters and Hak5) under our CDF-Zipf model are with a p-value> 0.01. Also note that, there is always the potential that more accurate Zipf models can be proposed to fit password distributions in the future, yet our CDF-Zipf model ensures that the max improvements in KS statistic D of such new models will be confined within [0, 0.018457]. That is, there is very limited room for improvement. Particularly, for 16 million Dodonew and 15 million 000webhost, the room for improvement is even less than 0.0062. All this suggests the accurateness of our CDF-Zipf model. The superiority of this model will be further illustrated in what follows.

D. Comparison of the Models

In an attempt to more accurately determine the Zipf distribution parameters, we have also employed a more complex approach proposed by Clauset *et al.* [44]. Their model was initially designed to characterize general Power-law distributions but not password distributions. It first needs to determine the parameters of a Power-law distribution, and then convert the Power-law parameters to the Zipf's law parameters. To accommodate discrete variables (e.g., the rank of a password in our setting), we employ [44, eq. 3.7] as suggested. We have employed their model to fit our 14 password datasets.[4] As shown in Table VII, the KS statistic D from fittings under Clauset et al.'s model [44] is 0.001079∼0.108263 (avg 0.031883), and hopefully four fittings are accepted by KS tests (i.e., with a p-value>0.01).

[4]We thank the source-codes http://tuvalu.santafe.edu/~aaronc/powerlaws/.

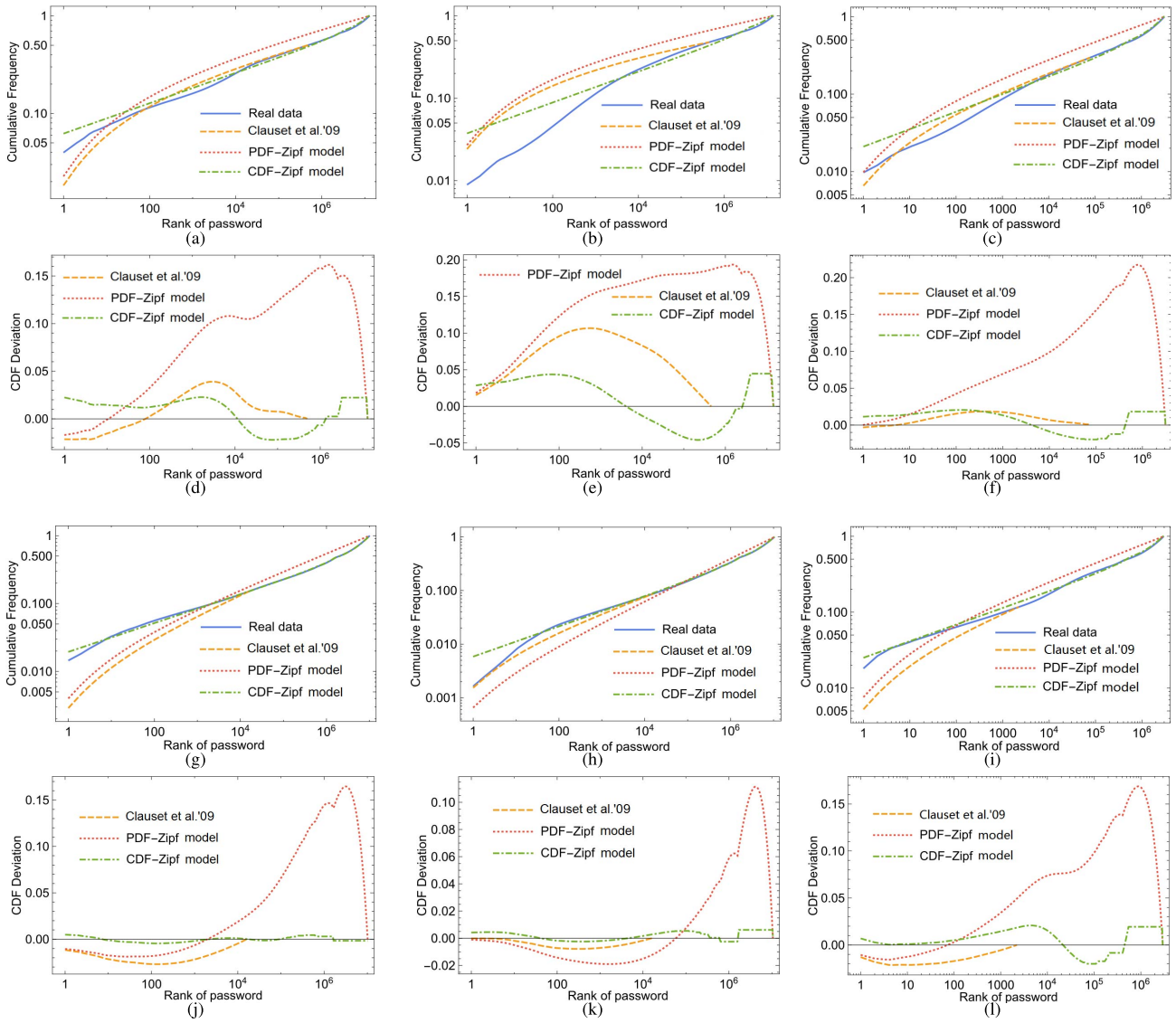


Fig. 3. A comparison of three different Zipf models, using 6 representative datasets for illustration. The absolute value of "CDF deviation" equals the KS statistic D. Our CDF-Zipf model performs much better than the other two: lower CDF deviations and 100% dataset coverage. See more details in Table VIII. (a) Approximating the distribution of Tianya. (b) Approximating the distribution of Rockyou. (c) Approximating the distribution of Gmail. (d) CDF deviation of each model when fitting Tinaya. (e) CDF deviation of each model when fitting Rockyou. (f) CDF deviation of each model when fitting Gmail. (g) Approximating the distribution of Dodonew. (h) Approximating the distribution of 000webhost. (i) Approximating the distribution of Mail.ru. (j) CDF deviation of each model when fitting Dodonew. (k) CDF deviation of each model when fitting 000webhost. (l) CDF deviation of each model when fitting Mail.ru.

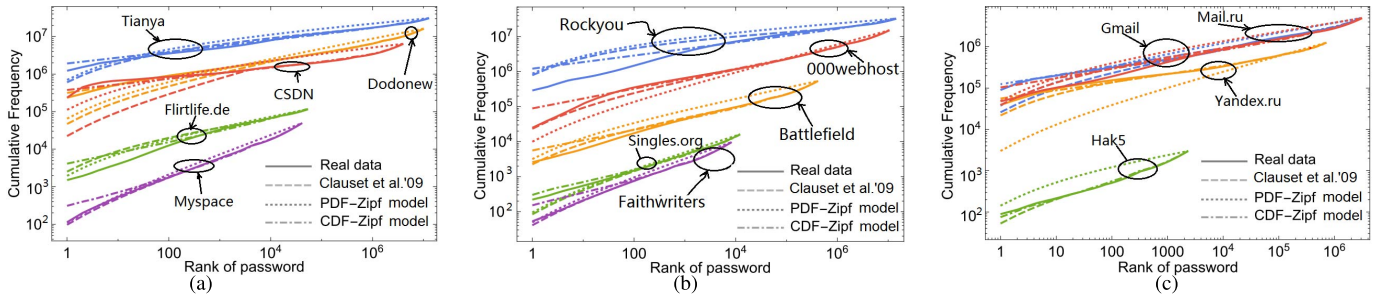


Fig. 4. Zipf's law in all 14 real-life password datasets from four varied languages, using 3 different fitting approaches. For detailed Zipf parameters from each model, see Tables IV, VI and VII. Generally, the superiority of these models are in the order: CDF-Zipf $\geq$ Clauset et al.'09 [44] $\geq$ PDF-Zipf. (a) Zipf's law in PWs of Chinese, English and German. (b) Zipf's law in PWs of English. (c) Zipf's law in PWs of Russian and English.

Table VIII provides a detailed comparison of the three Zipf models examined above. In terms of the KS statistic D, we can observe that: (1) The CDF-Zipf model and Clauset *et al.*'s model [44] are always better than the PDF-Zipf model; and (2) For million-sized datasets, the CDF-Zipf model generally performs the best among all,

TABLE VII
CLAUSET *et al.*'09 MODEL [44]: FITTING RESULTS OF 14 DATASETS

	x_{min}	Zipf s=$\frac{1}{\alpha-1}$	Dataset coverage	Statistic D	p-value
Tianya	5	2.099455	50.37%	0.046035	$<10^{-4}$
Dodonew	36	2.399279	15.15%	0.026957	$<10^{-4}$
CSDN	62	2.613033	20.52%	0.089842	$<10^{-4}$
Rockyou	5	2.061061	49.62%	0.108263	$<10^{-4}$
000webhost	23	2.423542	8.75%	0.007795	0.4160
Battlefield	5	2.378776	15.18%	0.008973	$<10^{-4}$
Flirtlife.de	4	2.233860	46.15%	0.064640	$<10^{-4}$
Myspace	7	2.964454	4.91%	0.001079	0.5080
Singles.org	4	2.885328	16.60%	0.009791	$<10^{-4}$
Faithwriters	4	2.999978	9.07%	0.001250	0.8368
Hak5	5	2.370435	9.41%	0.012166	0
Mail.ru	55	2.306956	11.72%	0.021279	0.8156
Gmail	9	2.215576	24.33%	0.025722	$<10^{-4}$
Yandex.ru	35	2.098622	17.18%	0.022568	$<10^{-4}$

TABLE VIII
COMPARISON OF THREE ZIPF MODELS FOR PASSWORDS

Dataset	KS statistic D			Dataset (distribution) coverage		
	PDF-Zipf	CDF-Zipf	Clauset [44]	PDF-Zipf	CDF-Zipf	Clauset [44]
Tianya	0.161718	**0.022798**	0.046035	52.86%	**100.00%**	50.37%
Dodonew	0.164640	**0.004926**	0.026957	29.92%	**100.00%**	15.15%
CSDN	0.268982	**0.022319**	0.089842	31.67%	**100.00%**	20.52%
Rockyou	0.193567	**0.045874**	0.108263	51.86%	**100.00%**	49.62%
000webhost	0.111546	**0.006170**	0.007795	23.09%	**100.00%**	8.75%
Battlefield	0.225527	0.010557	**0.008973**	17.14%	**100.00%**	15.18%
Mail.ru	0.168754	**0.020773**	0.021279	35.15%	**100.00%**	11.72%
Gmail	0.217463	**0.020543**	0.025722	32.05%	**100.00%**	24.33%
Flirtlife.de	0.062585	**0.036448**	0.064640	46.15%	**100.00%**	46.15%
Myspace	0.126651	0.008262	**0.001079**	9.66%	**100.00%**	4.91%
Singles.org	0.100741	0.013786	**0.009791**	16.60%	**100.00%**	16.60%
Faithwriters	0.140562	0.014524	**0.001250**	9.07%	**100.00%**	9.07%
Hak5	0.278761	0.018078	**0.012166**	11.28%	**100.00%**	9.41%
Yandex.ru	0.097279	**0.013345**	0.022568	37.18%	**100.00%**	17.18%
Average	0.165627	**0.018457**	0.031883	28.84%	**100.00%**	21.35%

*A value in bold indicates that the corresponding model is the best one.

while Clauset *et al.*'s model [44] performs the best when password datasets are in small size (e.g., less than 1 million). For better comprehension, see Fig. 4. In terms of the dataset coverage, we can observe that: in all cases, the CDF-Zipf model performs the best (i.e., achieving 100% coverage), the PDF-Zipf model performs the second best, while Clauset *et al.*'s model [44] performs the worst. In all, *our CDF-Zipf model generally produces the most desirable fittings among the three Zipf models*.

We also examined the time complexity of the three models when fitting a given dataset. More specifically, the time complexity of our PDF-Zipf model, our CDF-Zipf model and Clauset et al.'s model [44] is in $\mathcal{O}(|\mathcal{DS}|)$, $\mathcal{O}(|\mathcal{DS}| \cdot log|\mathcal{DS}|)$ and $\mathcal{O}(|\mathcal{DS}|)$, respectively. For a concrete example, on a moderate multi-core computer (i7-4790K 4.00GHz CPU and 16G RAM), when fitting the 32 million Rockyou passwords, our PDF-Zipf model takes 32.40 seconds, our CDF-Zipf model takes 14.67 hours, and Clauset et al.'s model [44] takes 69.39 seconds. In a nutshell, all three Zipf models can be completed in acceptable time with moderate computing resources.

E. General Applicability of our Zipf Models

The general applicability our Zipf models come from the diversity and wide representativeness of our 14 datasets. Section III shows that, our 14 datasets include passwords created before 2006 (see Myspace) and also as recent as Oct. 2015 (see 000webhost), cover 12 kinds of web services and four kinds of languages. They also represent a variety of culture (faith) backgrounds. Fortunately, results from both our PDF-Zipf model and CDF-Zipf model suggest that, these diversified datasets well follow the Zipf's law.

Particularly, among our 14 datasets, five kinds of different password policies can be inferred from Table II and III:

1) CSDN implements the policy "length $len \geq$8". Only 2.17% of passwords in CSDN are with $len <$8, while generally there are *10 times more* such short passwords in other Chinese sites. This implies that a transition in password policy has occurred: these 2.17% short passwords are created under the initial loose policy, while most of the 97.83% long passwords are created under the later enhanced policy (i.e., with $len \geq$8).
2) Myspace implements the policy "at least a letter and a number". Over 75.79%(=69.77%+6.02%) of passwords in Myspace are composed of both letters and numbers. In addition, 18.24% of users select passwords with a sequence of letters ended with "1", which is 4∼9 times higher than the other English sites. This highly indicates that there was a transition in password policy at sometime before the hacking happened, though by no means can we confirm this transition.
3) 000webhost implements the policy "at least a letter and a number, and $len \geq$6". Only 0.01% of passwords in 000webhost are with $len <$6, while generally there are *50 times more* such short passwords in other Chinese sites. This is likely due to a transition in password policy: these 0.01% short passwords are created under the initial loose policy, while most of the 99.99% long passwords are created under the later enhanced policy: $len \geq$6.
4) Singles.org implements the policy "$len \leq$8". 100% of passwords in Singles.org are with $len \leq$8, while generally such short passwords account for at most 80% in every other English sites. This is resulted from a obvious password policy: "$len \leq$8".
5) The other 10 sites show no apparent policy. They might implement no policy all along, or have conducted password policy transitions in the middle of their lifetime and thus the new password policies are not obvious from the resulting passwords.

As shown above, our password datasets are generated under a wide variety of password policies: from very loose ones to the restrictive ones "length $len \geq$8" and "at least a letter and a number, and $len \geq$6". Though currently we do not experiment on datasets that were generated under more restrictive policies (e.g., "at least a letter, a number and a symbol, and $len \geq$8"), one can have a high confidence that passwords created under such policies will also follow the Zipf's law. We leave the confirmation of this conjecture as an open issue.

In all, our datasets cover a wide variety of service types, sizes, how leaked, user localization, languages, faith, culture background and password policies. This demonstrates the wide applicability of our Zipf models and they can be used to well characterize the distribution of user-chosen passwords.

F. Some Implications

We now sketch three implications that our Zipf theory may have. For more details, readers are referred to [51].

1) For Password-Based Cryptographic Protocols: We propose to use the formulation $C' \cdot Q(k)^{s'}$ to capture an attacker's advantages in making at most $Q(k)$ on-line guesses against password-based cryptographic protocols, superseding the traditional ones (i.e., $Q(k)/|\mathcal{D}|$ [52], [53] and $Q(k)/2^m$ [54], [55]), where k is the system security parameter, $\mathcal{D}$ is the password space, C' and s' are the CDF-Zipf regression parameters of dataset $\mathcal{D}$, and m denotes the min-entropy of $\mathcal{D}$. Experiments on our 14 large-scale password lists show the superiority of our new formulation over existing ones. Generally, given a target system, the values of C' and s' can be approximated by leaked datasets from sites with a similar service, language (and policy). For instance, if the password protocol is to be deployed in a Chinese e-commerce site, one can set $C' = 0.019429$ and $s' = 0.211921$, which come from the Dodonew passwords (see Table VI).

2) For Password Creation Policies: Based on the Zipf assumption of passwords, we propose a series of prediction models to facilitate the choices of the threshold parameter $\mathcal{T}$ for the promising popularity-based password creation policy in [5]: (1) These passwords with a popularity above $\mathcal{T}$ account for the percentage $\eta = (\frac{\mathcal{T}}{C' \cdot s'})^{\frac{1}{s'-1}} \cdot (C')^{\frac{1}{s'}}$; (2) The percentage of users that will be potentially affected is $W_p(\eta) = \eta^{s'}$; and (3) The percentage of users that will be actually affected is $W_a(\eta) = (1 - s') \cdot \eta^{s'}$. Our models provide new insights and highlight that, usability will be largely impaired if $\mathcal{T}$ is improperly chosen. For instance, when setting $\mathcal{T} = 1/10^6$ (which is widely recommended [5], [24]) for Internet-scale sites, our model predicts that an average of 38.73% of users will be potentially annoyed. Our theory well accords with the extensive experiments.

3) For the α-Guesswork Metric: Under the Zipf assumption of password distribution, we reveal that the widely used password strength metric α-guesswork [20], which was believed to be always parametric with the success-rate α, is actually non-parametric in two of four cases. As passwords are generally Zipf-distributed, this result makes α-guesswork much simpler to use — now we only need a single value of the advantage α instead of "all values of α" [20] to inform decisions.

Summary: Different from the conclusion made in previous research (e.g., [19], [20]) that user passwords are "unlikely to actually be Zipf distributed", our models show that Zipf's law does exist in real-life passwords. The comparison results reveal that our CDF-Zipf model performs the best, while our PDF-Zipf model is worse than Clauset et al.'s model [44]. Our CDF-Zipf model is superior to the other two models in terms of both the KS statistic D and the dataset coverage. Particularly, Our CDF-Zipf model achieves remarkable accuracy: its max CDF deviation (i.e., the KS statistic D) is 0.006170∼0.045874 (avg 0.018457). To our knowledge, our datasets are so far among the most diversified and the largest ones in password studies, and they are of sound representativeness. It is expected that our Zipf theory would provide a much better understanding of the distributions of human-generated passwords, and it has already been adopted in other important password-related areas than what we have discussed (e.g., password encryption [56], password hash [25], password-datasets generation [57], password-cracking [3] and password manager [58]).

V. Strength Metric for Password Dataset

In this section, we address the question as to how to accurately measure the security 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 guessing attacker, 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], [26].[5] Accordingly, we can use the cracking result $\lambda^*_{\mathcal{X}}(n)$ to be the strength metric of a given dataset (distribution) $\mathcal{X}$:

$$\lambda^*_{\mathcal{X}}(n) = \sum_{r=1}^{n} p_r(\mathcal{X}) = \frac{1}{|\mathcal{DS}|} \sum_{r=1}^{n} f_r(\mathcal{X}), \tag{5}$$

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

In Section IV-C, we have shown that the distribution of passwords can be well approximated the law: $p_r(\mathcal{X}) = f_r(\mathcal{X})/|\mathcal{DS}| = F_r(\mathcal{X}) - F_{r-1}(\mathcal{X}) \approx C' \cdot r^{s'} - C' \cdot (r-1)^{s'}$. Consequently, $\lambda^*(n)$ is essentially determined by C' and s':

$$\lambda^*_{\mathcal{X}}(n) = \sum_{r=1}^{n} p_r(\mathcal{X}) = F_n(\mathcal{X}) \approx C' \cdot n^{s'} = \lambda_{\mathcal{X}}(n). \tag{6}$$

B. Evaluation

It should be noted that, in Eq. 6, $\lambda^*_{\mathcal{X}}(n)$ is not *exactly* equal to the value of the rightmost hand even though our CDF-Zipf model fits the actual data very well. We plot $\lambda^*_{\mathcal{X}}(n)$ as a function of n according to Eq. 5 and $\lambda_{\mathcal{X}}(n)$ as a function of n according to Eq. 6, and put these two curves together to see how they agree with each other. In Fig. 5(a), we depict $\lambda^*_{\mathcal{X}}(n)$ and $\lambda_{\mathcal{X}}(n)$ for 16 million passwords from the Dodonew dataset and obtain an max deviation of 0.49% (avg 0.19%) for the two curves. Due to space constraints, here we cannot illustrate the related pictures for the other datasets like that of Dodonew and 000webhost, yet we summarize the average deviation between the two curves $\lambda^*_{\mathcal{X}}(n)$ and $\lambda_{\mathcal{X}}(n)$ ($1 \leq n \leq |\mathcal{DS}|$) for each dataset in Table IX.

Table IX shows that, except for Rockyou and Flirtlife.de, the average deviations are all below 2% (i.e., from 0.19% to 1.81%), suggesting sound consistence of $\lambda_{\mathcal{X}}(n)$ with the optimal attacking result $\lambda^*_{\mathcal{X}}(n)$. This means that the $\lambda^*_{\mathcal{X}}(n)$

[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.

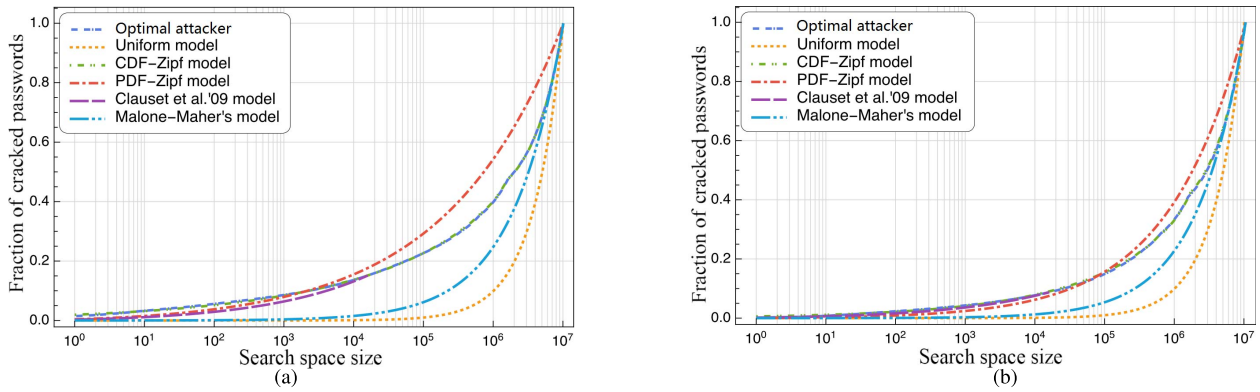


Fig. 5. Consistence of optimal attack with our CDF-Zipf-based metric (i.e., $\lambda_{\mathcal{X}}(n)$) on two example datasets (16.2M Dodonew and 15.3M 000webhost). (a) Approximating the optimal attacker against Dodonew. (b) Approximating the optimal attacker against 000webhost.

TABLE IX
THE DEVIATION BETWEEN $\lambda^*_{\mathcal{X}}(n)$ AND $\lambda_{\mathcal{X}}(n)$ $(1 \leq n \leq |\mathcal{DS}|)$

Dataset	Max deviation	Avg deviation
Tianya	2.28%	1.81%
Dodonew	0.49%	0.19%
CSDN	2.23%	1.15%
Rockyou	4.59%	3.57%
000webhost	0.62%	0.54%
Battlefield	1.06%	0.65%
Mail.ru	2.08%	1.75%
Gmail	2.05%	1.70%
Flirtlife.de	3.64%	2.83%
Myspace	0.83%	0.93%
Singles.org	1.38%	0.90%
Faithwriters	1.45%	0.99%
Hak5	1.81%	0.79%
Yandex.ru	1.33%	1.05%
Average	**1.85%**	**1.35%**

curve well overlaps with the $\lambda_{\mathcal{X}}(n)$ curve for each dataset. As shown in Fig. 5, these two curves for both datasets almost overlaps with each other. We also provide a concrete comparison of our CDF-Zipf model with the other four models in terms of how well they approximate the real attacker. Fig. 5 shows that our CDF-Zipf based metric $\lambda_{\mathcal{X}}(n)$ performs the best, followed by Clauset et al.'s model based metric.

Now that the optimal attack can be well approximated by $\lambda_{\mathcal{X}}(n)$, it is natural to propose $\lambda_{\mathcal{X}}(n)$ to be the metric for measuring the strength of password dataset $\mathcal{X}$, where n is the number of guessing attempts.

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 specially, by introducing a number of NLP techniques and statistic-based computational theories, we propose two Zipf-like models to characterize the distribution of passwords: *PDF-Zipf* and *CDF-Zipf*. Extensive experiments based on fourteen large-scale datasets, which consist of 113.3 million real-world passwords, show that our PDF-Zipf model can well fit the popular passwords (i.e., with $f_r \geq 4$) and obtain $R^2 > 0.97$, while our CDF-Zipf model can well fit the entire password dataset, with the maximum CDF deviation of the empirical distribution and the fitted theoretical model being 0.48%∼4.59% (avg. 1.84%).

In comparison, our CDF-Zipf model not only covers 100% of a given password dataset, but also is generally more accurate than both Clauset et al.'s model [44] and our PDF-Zipf model. Thus, we recommend the use of our CDF-Zipf model to characterize password distributions. However, two out of our fourteen datasets (i.e., Rockyou and Flirtlife.de) are with the maximum CDF deviation >3% and the KS test p-value $<10^{-4}$. This suggests the limitation of our CDF-Zipf model: there is no single distribution function that can perfectly fit all kinds of password distributions. In other words, our CDF-Zipf model can be further tuned to cater for some cases, e.g., extending our power law $F_r = C' \cdot r^{s'}$ to a power law with exponential cutoff $F_r = C' \cdot r^{s'} \cdot \theta^r$ or a shifted power law $F_r = C' \cdot (r+\theta)^{s'}$, where C', s' and θ are constants. Still, our CDF-Zipf model is very accurate for most of the datasets (i.e., with the maximum CDF deviations being 0.49% ∼2.28% and their *average* being 1.85%), and even for Rockyou and Flirtlife.de, it obtains 2 times better fittings (in terms of KS statistic D) than other existing models.

Armed with the concrete distribution function of passwords, we suggest a new metric for measuring the strength of password creation policies. We further briefly sketch three important applications of our Zipf theory. 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. For instance, our Zipf theory and numerical results have already been adopted in a wide range of password-related domains such as password encryption [56], password hash [25], password-dataset generation [57] and password manager [58].

More work remains to be done on this interesting yet challenging topic. For instance, how to more accurately (and efficiently) determine the Zipf parameters of password distributions? What is the underlying mechanism that leads to the emergence of Zipf's law in passwords? How will the password distribution of a system evolve as time goes on? Do passwords generated under a much restrictive policy (e.g., "at least a letter, a number and a symbol, and length≥8")

obey Zipf's law? Do extremely high value passwords (e.g., for e-banking accounts) follow Zipf's law? This highlights the need for more attention from a wide range of research domains to join forces to address these issues. It is also a mixed blessing that, the chances for such investigations to be conducted in the future are increasing as more sites of high values are breached and more password datasets (and the associated user behavior information) are made publicly available.

ACKNOWLEDGMENT

The authors are grateful to the anonymous reviewers for their invaluable comments that improve the completeness and readability of this paper. A preliminary version of this work appeared online at https://eprint.iacr.org/2014/631.

REFERENCES

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

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

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

[4] J. H. Huh, S. Oh, H. Kim, and, "Surpass: System-initiated user-replaceable passwords," in *Proc. CCS*, 2015, pp. 170–181.

[5] 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.

[6] W. Burr, D. Dodson, R. Perlner, S. Gupta, and E. Nabbus, "Electronic authentication guideline," Nat. Inst. Standards Technol., Reston, VA, USA, Tech. Rep. NIST SP800-63-2, Aug. 2013.

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

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

[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 *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 *et al.*, "Measuring password guessability for an entire University," in *Proc. ACM CCS*, 2013, pp. 173–186.

[13] B. Ur *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 Security Privacy*, vol. 11, no. 5, pp. 80–83, Sep. 2013.

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

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

[17] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Device-enhanced password protocols with optimal online-offline protection," in *Proc. ASIACCS*, 2016, pp. 177–188.

[18] Z. Zhang, K. Yang, X. Hu, and Y. Wang, "Practical anonymous password authentication and TLS with anonymous client authentication," in *Proc. ACM CCS*, 2016, pp. 1179–1191.

[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 SP*, May 2012, pp. 538–552.

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

[22] E. H. Spafford, "OPUS: Preventing weak password choices," *Comput. Secur.*, vol. 11, no. 3, pp. 273–278, 1992.

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

[24] D. Florêncio, C. Herley, and P. van Oorschot, "An administrator's guide to Internet password research," in *Proc. USENIX LISA*, 2014, pp. 44–61.

[25] J. Blocki and A. Datta, "CASH: A cost asymmetric secure hash algorithm for optimal password protection," in *Proc. IEEE CSF*, Sep. 2016, pp. 1–10. [Online]. Available: http://arxiv.org/pdf/1509.00239v1.pdf

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

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

[28] R. Bowes. (May 2015). *Passwords*. [Online]. Available: https://wiki.skullsecurity.org/Passwords

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

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

[31] S. Designer. (Feb. 1996). *John Ripper Password Cracker*. [Online]. Available: http://www.openwall.com/john/

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

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

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

[35] R. Martin. (Dec. 2011). *Amid Widespread Data Breaches China*. [Online]. Available: http://www.techinasia.com/alipay-hack/

[36] L. Constantin. (Jul. 2009). *Security Gurus 0wned by Black Hats*. [Online]. Available: http://www.programdoc.com/1114_98794_1.htm

[37] C. Allan. (Dec. 2009). *32 Million Rockyou Passwords Stolen*. [Online]. Available: http://www.hardwareheaven.com/news.php?newsid=526

[38] A. Dobra. (May 2011). *Hacker Group LulzSec Leaks Battlefield Heroes Accounts, Has Now Retired*. [Online]. Available: http://news.softpedia.com/news/Hacker-Group-LulzSec-Leaks-Battlefield-Heroes-Accounts-Has-Now-Retired-208172.shtml

[39] J. Mick. (Sep. 2014). *Russian Hackers Compile List 10M+ Stolen Gmail, Yandex, Mailru*. [Online]. Available: http://t.cn/R4tmJE3

[40] B. Schneier. (May 2006). *Password Data Flirtlife. de Compromises*. [Online]. Available: https://www.schneier.com/blog/archives/2006/05/common_password.html

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

[42] C. Johnston. (Apr. 2013). *Why Your Password Cant Have Symbols*. [Online]. Available: http://t.cn/zTNoEEs

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

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

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

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

[47] 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.

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

[49] S. Nolan, *Study Guide for Essentials of Statistics for the Behavioral Sciences*. London, U.K.: Worth Publishers, 2013.

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

[51] D. Wang and P. Wang, "On the implications of Zipf's law in passwords," in *Proc. ESORICS*, 2016, pp. 111–131.

[52] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the J-PAKE password-authenticated key exchange protocol," in *Proc. IEEE SP*, May 2015, pp. 571–587.

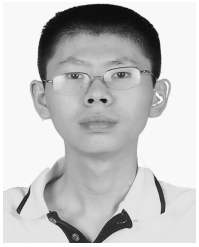
[53] J. Katz, R. Ostrovsky, and M. Yung, "Efficient and secure authenticated key exchange using weak passwords," *J. ACM*, vol. 57, no. 1, pp. 1–41, 2009.

[54] F. Kiefer and M. Manulis, "Zero-knowledge password policy checks and verifier-based PAKE," in *Proc. ESORICS*, 2014, pp. 295–312.

[55] M. Abdalla, F. Benhamouda, and D. Pointcheval, "Public-key encryption indistinguishable under plaintext-checkable attacks," in *Proc. PKC*, 2015, pp. 332–352.

[56] Z. Huang, E. Ayday, J. Fellay, J.-P. Hubaux, and A. Juels, "Genoguard: Protecting genomic data against brute-force attacks," in *Proc. IEEE SP*, May 2015, pp. 447–462.
[57] W. Han, L. Yuan, S. Li, and X. Wang, "An efficient algorithm to generate password sets based on samples," *Chin. J. Comput.*, vol. 40, no. 5, pp. 1151–1157, 2017.
[58] M. Fukumitsu, S. Hasegawa, J.-Y. Iwazaki, M. Sakai, and D. Takahashi, "A proposal of a password manager satisfying security and usability by using the secret sharing and a personal server," in *Proc. IEEE AINA*, Mar. 2016, pp. 661–668.

Ding Wang received the Ph.D. degree in information security from Peking University in 2017. He is currently supported by the Boya Post-Doctoral Fellowship in Peking University, China. He has authored over 40 papers at venues like ACM CCS and IEEE TDSC, and his papers get over 800 citations. His research interests mainly focus on password-based authentication and provable security. He received the Top-10 Distinguished Graduate Academic Award from Harbin Engineering University in 2013 and Peking University in 2016.



Haibo Cheng received the B.S. degree in pure mathematics from Nankai University, China, in 2011, and the M.S. degree in pure mathematics from Peking University, China, in 2015. He is currently pursuing the Ph.D. degree in information security with Peking university. His work appeared at IFIP DSN 2016 and ACM ASIACCS 2016. His research interests focus on password-based authentication.

Ping Wang received the Ph.D. degree in computer science from the University of Massachusetts Amherst, USA, in 1996. He is currently a Professor with the National Engineering Research Center for Software Engineering, and the School of Software and Microelectronics, Peking University, China. He is the Director of Intelligent Computing and Sensing Laboratory, Peking University. He has authored over 50 papers in journals or proceedings, such as IEEE TDSC, ACM CCS, IEEE ICWS, and IEEE CloudCom. His research interests include information security and distributed computing.

Xinyi Huang received the Ph.D. degree from the University of Wollongong, Australia. He is currently a Professor with the School of Mathematics and Computer Science, Fujian Normal University, China, and a Co-Director of the Fujian Provincial Key Laboratory of Network Security and Cryptology. He has authored over 100 papers in refereed international conferences and journals. His research interests include applied cryptography and network security. He is an Associate Editor of IEEE TDSC.

Gaopeng Jian received the Ph.D. degree in pure mathematics from Peking University, China. He is currently a Post-Doctoral Researcher with Peking University, China. His research interests include coding theory and deep neuron networks.