Hindawi
Security and Communication Networks
Volume 2021, Article ID 5563884, 16 pages
https://doi.org/10.1155/2021/5563884

Research Article

Understanding Offline Password-Cracking Methods: A Large-Scale Empirical Study

Ruixin Shi,[1,2] **Yongbin Zhou**,[1,2] **Yong Li**,[3] **and Weili Han**[4]

[1] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[2] *School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[3] *Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
[4] *Software School, and Shanghai Key Laboratory of Data Science, Fudan University, Shanghai, China*

Correspondence should be addressed to Yongbin Zhou; zhouyongbin@iie.ac.cn

Received 19 January 2021; Revised 5 March 2021; Accepted 15 April 2021; Published 27 April 2021

Academic Editor: Salvatore D'Antonio

Researchers proposed several data-driven methods to efficiently guess user-chosen passwords for password strength metering or password recovery in the past decades. However, these methods are usually evaluated under ad hoc scenarios with limited data sets. Thus, this motivates us to conduct a systematic and comparative investigation with a very large-scale data corpus for such state-of-the-art cracking methods. In this paper, we present the large-scale empirical study on password-cracking methods proposed by the academic community since 2005, leveraging about 220 million plaintext passwords leaked from 12 popular websites during the past decade. Specifically, we conduct our empirical evaluation in two cracking scenarios, i.e., cracking under extensive-knowledge and limited-knowledge. The evaluation concludes that no cracking method may outperform others from all aspects in these offline scenarios. The actual cracking performance is determined by multiple factors, including the underlying model principle along with dataset attributes such as length and structure characteristics. Then, we perform further evaluation by analyzing the set of cracked passwords in each targeting dataset. We get some interesting observations that make sense of many cracking behaviors and come up with some suggestions on how to choose a more effective password-cracking method under these two offline cracking scenarios.

1. Introduction

Because of some irreplaceable advantages, such as low technical requirements and wide usage, textual passwords are likely to remain the most common authentication method for the near future [1]. Inevitably, there is a security-usability dilemma in textual passwords. Strong passwords are always hard to remember, so it is not surprising that users often create easy-to-guess passwords for convenience, which puts password-based authentication systems in a high-risk situation [2, 3]. Considering various attacks, offline cracking poses a serious threat and cannot be easily ignored [4]. This attack can be entirely performed under the attacker's control. The attacker can make as many attempts as possible to recover plaintext passwords from target hashed datasets given enough computational power. Unfortunately, due to frequent password leakage incidents [5–7], the security risk caused by this attack is exacerbated. Consequently, it is essential for password-based authentication systems to evaluate their resilience to offline cracking properly.

Compared with the traditional brute-force attack that is exhaustively trying all the possible character combinations [8], state-of-the-art password-cracking methods have significant advantages. They aim to simulate real-world cracking scenarios using leaked passwords to construct complex candidate passwords, which can expect to cover as many target passwords as possible while minimizing the number of trying. In this way, these methods have become a more promising mainstream metric for offline cracking [9–11]. Also, a sufficiently precise estimation of a password-based authentication system's ability to resist the most powerful offline attacks can only be provided when choosing the right cracking method. Therefore, the key to safeguard password-based authentication systems against offline attacks depends highly on selecting appropriate password-cracking methods.

Since 2005, several cracking methods have been proposed in academic research, which are quite different in many aspects. Only very few empirical research studies can be found in the literature regarding the comprehensive examination of the extant mainstream password-cracking methods. Some of these studies have only evaluated the performance of specific cracking methods [10, 12–16]. Meanwhile, each study was conducted under different settings, including various datasets and ad hoc cracking scenarios, making the experiment results disparate and inconsistent. Besides, some evaluation was limited by the lack of abundant plaintext passwords [17]. Also, the emergence of new cracking methods such as neural network-based methods [15, 18] was recently proposed and has not been thoroughly evaluated. There is lack of systematic comparison with other approaches, and these factors make empirical studies of password-cracking methods vary greatly. In this way, it causes confusion in understanding cracking methods and difficulty in comparing empirical results accurately and fairly. Therefore, it is necessary and meaningful to uniformly study these methods and elaborate on selecting the right method.

Thus, this motivates us to perform a systematic and comparative investigation with a very large-scale data corpus for such state-of-the-art offline cracking methods in order to address the problem of how attackers can choose a more effective offline password-cracking method and make our empirical results become a basis for fair and impartial estimation on the ability of the password-based authentication system to resist the most potent offline attacks. Aiming at this, we conduct a large-scale empirical study on password-cracking methods, including the latest one based on the neural network, leveraging about 220 million plaintext passwords leaked from 12 popular websites during the past decade.

As far as we know, Ji et al.'s work [19] may be the closest to this paper, but our work differs from his in several aspects. First, we perform more extensive experiments. According to the analysis of plaintext datasets characteristics, we define two cracking scenarios simulating real-world practice, including cracking under extensive-knowledge and limited-knowledge. From the perspective of datasets and methods, we use a larger number of plaintext datasets for evaluation. Also, we explore emerging cracking methods not covered in [19] such as LSTM- and GAN-based methods and new versions or settings of other methods. Second, we conduct further analysis about cracking efficiency in each scenario using a new measurement, which calculates the percentage of each type of password in the cracked dataset accounts for the corresponding subset of passwords in the targeting dataset.

We summarize our main contributions and some findings as follows:

(i) We perform a large-scale empirical study on 7 mainstream offline password-cracking methods proposed by the academic community since 2005, leveraging about 220 million passwords leaked from 12 popular websites in two scenarios, including cracking passwords *under extensive-knowledge* (the attacker knows exact password distribution of the target and can crack utilizing passwords that have the same distribution) and cracking *under limited-knowledge* (the attacker only knows regional information of the target and can only use passwords from a different source). Through a comprehensive analysis of the results, it is concluded that no cracking method may outperform others from all aspects in these two offline scenarios, and the actual cracking performance is determined by multiple factors, including the underlying model principle along with dataset attributes such as length and structure characteristics.

(ii) We conduct further evaluation by analyzing the set of cracked passwords in each targeting dataset, and we got some interesting observations. One essential finding is that attackers can increase cracking efficiency by analyzing the characteristics of password datasets. By observing experiment results, particular attention factors include password length distribution and structure composition, and regional or language information of the training and targeting datasets. For example, we found that the Markov based method is more suitable for cracking Chinese datasets, while the PCFG-based method works better on English datasets; Two neural network-based methods demonstrate totally opposite effect, FLA performs surprisingly well, but GAN shows badly; The rule-based method is unstable and can be used as a complement to others; region information does strongly affect the choice of cracking methods. These results make sense of many cracking behaviors and can be used as the basis for how to choose a more effective offline cracking method.

This paper is organized as follows: Section 2 introduces related work. In Section 3, we present the datasets and cracking methods under our evaluation. Section 4 performs a large-scale empirical study on the cracking methods. We discuss the limitations and future work in Section 5, and Section 6 summarizes our work and shows our conclusions.

2. Related Work

2.1. Password-Cracking Methods. Password offline attacks can reach an unlimited number of cracking attempts given enough computational power [16]. It is crucial to properly evaluate the ability of password-based authentication systems resilience to this attack, according to [4]. Compared to exhaustively trying all possible strings, password-cracking methods can construct dictionaries that are more in line with human password creation behavior, which makes them become a more promising choice for offline cracking [9–11]. During the past decades, various cracking methods have been proposed. The authors in [20] introduced the

Markov-based model into password cracking. Ma et al. [13] improved this method by using normalization and smoothing to solve the overfitting phenomenon in a high-ordered model. Dürmuth et al. [14] implemented an ordered Markov-based password guessing method named Ordered Markov ENumerator (OMEN), which generates candidate passwords in decreasing order of possibility. Weir et al. [12] proposed a probabilistic model for password cracking called probabilistic context-free grammars (PCFGs) using the idea of applying context-free grammar into password structures. In later studies, semantic patterns were viewed as segments inserted into the dictionary to improve efficiency of PCFG [21–23]. In [24], they studied the long passwords and proposed a framework TransPCFG, which transfers the knowledge from short passwords to facilitate long password guessing. With the development of deep learning in text generation, Melicher et al. [18] used the recurrent neural network to build a password-cracking model in 2016, which introduces LSTM in the domain of passwords for the first time. Results of [18] show the good potential that LSTM can outperform traditional methods when evaluating passwords with the structure of 1class8 and 3class12. Hitaj et al. [15] proposed another password-cracking method using the generative adversarial network (GAN), which uses an adversarial process to generate passwords. Other than these, there are also many commercial cracking tools such as Hashcat [25] and John the Ripper [26], which support multiple modes, such as dictionary mode and mask attack mode.

2.2. Empirical Studies on Cracking Methods. There are several empirical studies involved in password-cracking methods. In 2010, Dell'Amico et al. conducted an empirical analysis of the password strength of 58,800 users [16]. However, only a few password-cracking methods developed before 2010 were studied then. The latest related work is [19], where Ji et al. evaluated the vulnerability of current password systems against password-cracking algorithms in 2017. However, only PCFG schemes, Markov model schemes, and two password crackers were evaluated. We are also inspired by [21, 27], which performed extensive, empirical analysis of real-world Chinese website passwords and English counterparts. These works provided a quantitative measurement of to what extent their native language influences user passwords. In addition, Mori et al. [28] determined the propensity of password creation through the lens of three language spheres. Results in [10] showed that configuration affects cracking efficiency a lot. It is not reliable using only one cracking method to measure password strength, which concluded that automated guessing methods could often approximate professionals' guessing.

Other than these, the authors in [29–31] studied password reuse and expiration practice. Mazurek et al. [17] measured password guessability for an entire university. In [13], Ma et al. performed experiments on probabilistic password models. Ji et al. [32] investigated an empirical study on password correlation quantification and application. Ye et al. [33] conducted an online empirical study to evaluate four kinds of mnemonic password creation tips. Moreover, Zeng et al. [34] studied lexical sentiment in passwords from Chinese websites. Walia et al. [35] examined a dataset of more than 7 million passwords to determine whether the user generated passwords are secure and established a relationship between the two.

3. Methodology

In this section, we introduce password datasets and cracking methods under our evaluation.

3.1. Datasets

3.1.1. Basic Information of Datasets. Similar to other tasks of empirical analysis, a large number of datasets are the precondition for evaluating [13, 16, 21, 22]. In this paper, we collect 12 leaked password datasets, which consist of about 220 million real-world passwords and are used in various computer applications and systems (e.g., email, gaming, and social forum). They were leaked due to various password leakage incidents between 2009 and 2015 [5–7], and all of them are publicly available now. Detailed taxonomies of them are shown in Table 1. According to [21, 27, 36], these datasets are from two different language domains. Most users of 163, Duduniu, 178, CSDN, Sinaweibo, and Duowan are Chinese speaking users, and the rest are English-speaking users. We refer to them as Chinese password datasets and English counterparts, respectively. These websites are all widely used and have many users, so it is reasonable for these datasets to represent current human password creation behaviors. From this point, these datasets can comprehensively characterize the statistical distribution of user-created passwords, which helps evaluate password-cracking methods.

(1) Ethical Discussion. Here, we declare our ethical considerations. Till now, these datasets have already been widely used for meaningful academic research [13, 16, 21, 22] and made a positive effect on password security. In this paper, we exclude personally identifiable information such as user names, emails, and only use passwords for research purposes. Therefore, we followed the ethical practice and would not cause additional harm to the victims.

3.1.2. Dataset Analysis. We first launch statistical analysis on the datasets above in terms of length, structure, and strength. These characteristics are of particular interest since they are most frequently used in password composition strategies and previous studies.

(1) Length Distribution. From Figure 1, we observe that the majority of total datasets have password length between 6 and 15, accounting for 96.27% of all lengths, and this proportion can reach 99% in some datasets. Very few users prefer passwords longer than 15 to memorize difficulty probably. Note that each password dataset has its specific length distribution due to creation rules specified by the

Table 1: Basic information of password datasets.

Dataset	Language	Web service	Leaked year	Password number
163	Chinese	Email	2015	116,503,433
Duduniu	Chinese	Internet service	2011	15,987,361
178	Chinese	Gaming	2011	9,049,058
CSDN	Chinese	Programmer forum	2011	6,413,632
Sinaweibo	Chinese	Social forum	2011	4,550,235
Duowan	Chinese	Gaming	2011	3,908,495
Rockyou	English	Social forum	2009	32,585,350
000Webhost	English	Web hosting	2015	15,203,342
Xato	English	Blog	2015	9,997,887
Gmail	English	Email	2014	4,800,930
Yahoo	English	Web portal	2012	442,836
Phpbb	English	Programmer forum	2009	255,376
Total number				219,697,935

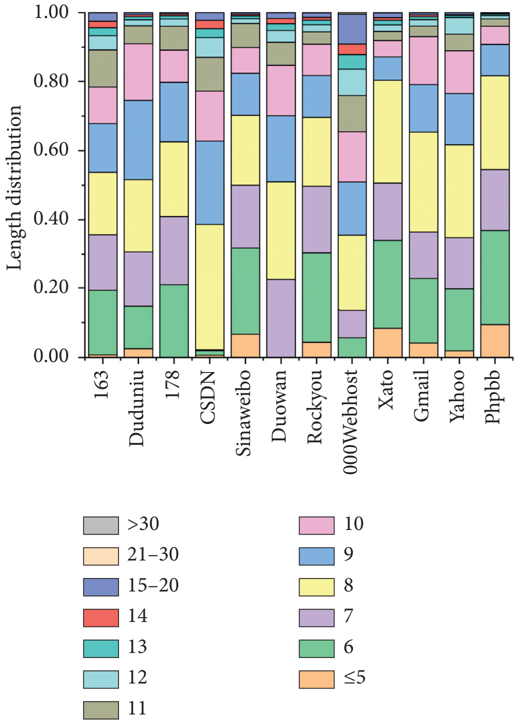

Figure 1: Length distribution.

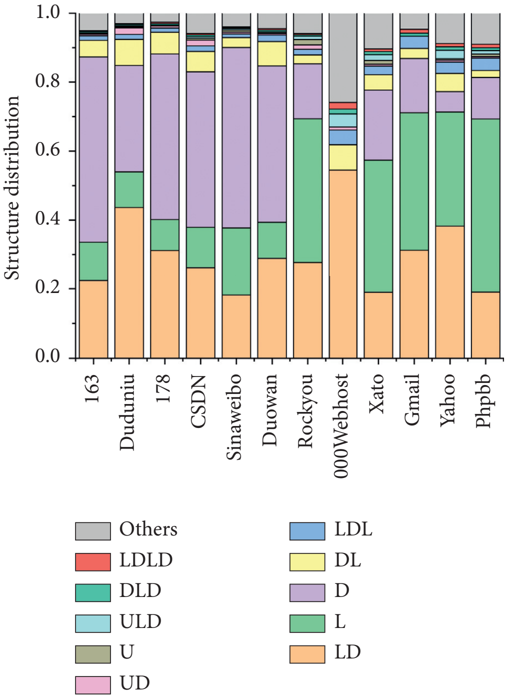

Figure 2: Structure distribution.

website. However, there is a similar distribution feature between Chinese and English password datasets, although users in different languages created these two groups.

(2) Popular Structures. We use representations in the probabilistic context-free grammar [12] (U = uppercase, L = lowercase, D = digit, and S = symbol). For example, the structure of abc123K! is modeled as LDUS. Figure 2 depicts the top 10 popular structures derived from [13, 19], which shows the most common structure in Chinese and English password datasets are L, LD, and D, and these structures account for most of the proportion. Unlike length distribution, it is worth noting that datasets from different language groups or regions have significantly varied structure distributions. From Figure 2, pattern L is the most popular structure in English datasets (000Webhost requires users to create passwords with more than one type of character), followed by LD. While in Chinese datasets, D is the most common one and makes up a larger part (about 30%–60%) of Chinese password datasets. Despite creation in a very diverse range of website types, passwords from the same language group or regions have quite similar structure distributions. It has to do with users creating passwords based on their language habits. Chinese users are not native English speakers, so digits appear to be the best password candidate. This phenomenon suggests that one can largely determine its users' native language when given a password dataset by investigating its structure distribution. The impact of language or regional differences between the two dataset groups will be discussed in the following sections.

(3) Password Strength. During password creation, strength meters embedded in a registration page can instantly evaluate and output the strength of given passwords [37], guiding users to choose passwords correctly. Here, we use a commercial password strength meter Zxcvbn [38], an open-sourced client-side password strength checker developed by Dropbox [39]. As evident from [40], Dropbox's relatively simple checker effectively analyzes passwords. Not surprisingly, from Figure 3, Zxcvbn classifies most passwords (more than 50%) of each dataset as either very weak or weak even though it has five levels. Besides, there is no significant difference in password strength between Chinese and English users. Among 12 datasets, 000Webhost has more secure passwords than others. This implies that the Zxcvbn meter is cautiously, and users always choose weak passwords for some reason. Therefore, it is essential to evaluate the ability of passwords to resist offline cracking effectively.

(4) Data Cleaning. When statistically analyzing these original datasets, we note that some of the datasets contain unnecessary information such as strings with length >100 and descriptions. Thus, before any evaluation, we first perform data cleaning. We reserve the legal passwords containing only 95 printable characters and further remove the passwords of a length less than 6 or greater than 30 [13, 27] because we find there is a good chance that these long strings are useless information when looking carefully and beyond the attacker's concern about cracking efficiency [41]. Other than that, passwords with lengths less than 6 do not satisfy most websites' creation strategy [40, 42]. Based on the analyzing results above, the proportion of filtered passwords is negligible.

3.2. Cracking Methods. We investigate 7 state-of-the-art mainstream password-cracking methods, which can be classified into four categories: rule-based methods, Markov-based methods, probabilistic context-free grammar-based methods, and neural network-based methods. They are selected based on the popularity in the literature, as well as their conceptual distinctness.

3.2.1. Rule-Based Methods. The rule-based method is a popular strategy in cracking and is widely used in commercial password-cracking tools, both Hashcat [25] and John the Ripper [26]. The main idea of rule-based methods is combining training lists with mangling rules and transforming original passwords into new candidate passwords. Typical rules include appending characters, reversing the items, capitalizing the first letter, and so on. Although these rules work well in practice, creating and expanding new rules is a labor-intensive task requiring specialized expertise. This method generates passwords with all the same probability without calculating the probability, so it is one of the fastest ways to get many candidate passwords. Here, we choose the Hashcat Best64 ruleset (denoted as Best64) for evaluation because it is widely used in real cracking activities and consistent with security research best practices seen in [15].

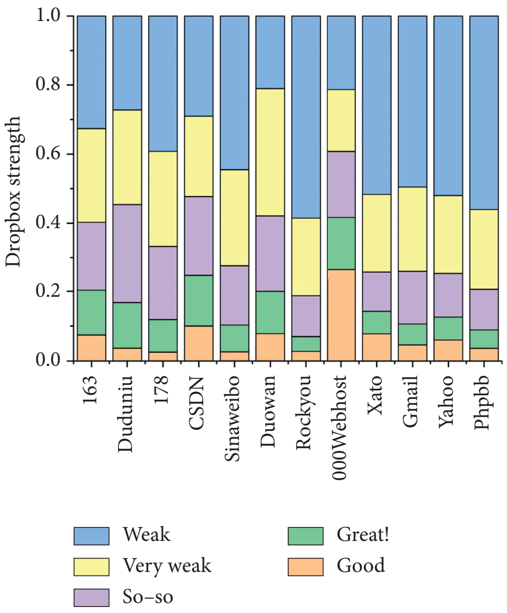

FIGURE 3: Dropbox strength.

3.2.2. Markov-Based Methods. The core assumption is that users construct passwords from front to back. After training the whole password, one can calculate the password's probability through the connection between characters from left to right. This method is divided into two stages. During training, the n-gram model was trained, and the frequency of each letter appeared after the substring of length n is counted. The training time for this method is short. During the generation stage, the probability of a probable password is calculated according to the Markov-chain, and then the candidate passwords are generated. For example, the probability of "mark" in the 4-gram Markov-based model is below, in which $\wedge$ and $\vee$ are the start and end symbol, respectively:

$$\begin{aligned}\Pr(\text{mark}) &= \Pr(m|\wedge) * \Pr(a|m) * \Pr(r|ma) * \Pr(k|\text{mar}) * \Pr(\vee|\text{mark}),\\ \Pr(r|ma) &= \frac{\text{Count}(r|ma)}{\sum_{\alpha\in\Gamma}\text{Count}(\alpha|ma)}.\end{aligned}\tag{1}$$

There are three parameters n-gram size, alphabet size, and the number of levels for enumerating passwords in this method. n-gram has the most significant impact on accuracy [14]. A larger n-gram usually provides a more accurate approximation of password distribution. However, this needs longer runtime, as well as larger memory requirements. When the order is too high, it will cause many strings to appear as 0. Ma et al. [13] introduced smoothing techniques to solve the data sparsity situation. We implement OMEN of [14] and set the order of the Markov model between 2 and 5 with an alphabet size of 72, and 10 levels based on [14]. We also add Laplace smoothing of all orders, and the value of θ is 0.01 as in [13]. In our experiments, 2-gram and 3-gram did not perform well, so those are not mentioned in the following. We select two best models: the 4-gram model with Laplace (denoted as OMEN) and the 5-gram model with Laplace (denoted as OMEN-5) for the next evaluation.

3.2.3. Probabilistic Context-Free Grammar-Based Methods. The core assumption of this method is that password segments are independent of each other. This method is inspired by the idea of analyzing the grammatical structure of statements in NLP, which regards password structures as grammars and divides passwords into different segment types according to their character composition. When performing an attack, one needs to choose a training set to extract grammars and structure frequency. The specific substring frequency in the corresponding segment is counted to calculate the probability of probable passwords. Then, candidate passwords are generated in descending order according to frequency to simulate passwords' probability distribution in reality. For example, the structure of "PCFG123!#" is "L4D3S2." The number after each substring indicates the segment's length, and the probability of "PCFG123!#" is shown as follows:

$$\begin{aligned}\Pr(\text{PCFG123!\#}) = \Pr(L4\ D3S2) * \Pr(\text{PCFG}|L4)\\ * \Pr(123|D3) * \Pr(!\#|S2).\end{aligned} \tag{2}$$

We use two open-sourced versions of this method developed by Matt Weir in our later experiments. The original [12, 43] (denoted as PCFG) is of general purpose, but this one gets slower through generating, and it needs 2-3 days to generate enough candidate passwords. We also evaluate the latest version of PCFG [44] (denoted as PCFG-4), which adds richer structures such as keyboard and uppercase and integrates with the Markov model into the grammar. The parameter is set as default. We regard the training dataset as the dictionary file to generate passwords based on the conclusion in [13] to improve efficiency.

3.2.4. Neural Network-Based Methods. Deep learning models are recently used to construct passwords and have become a new direction in password cracking. We divide deep learning-based methods into probabilistic models that generate candidate passwords according to their probability and generative models that randomly generate candidates in batches.

In 2016, Melicher et al. used LSTM to construct a password guessing model (denoted as FLA) [18]. FLA is similar to Markov based methods for they both calculated the probability of probable password by predicting the next character and its probability after fixed-length substrings. However, different from Markov, FLA does not need to manually count and record the frequency of appeared characters after each substring in the training set. The next most likely character and its probability will output by inputting a fixed-length substring to a well-trained neural network. This method is based on the probabilistic model and has shown its advantages in modeling password guessability through Monte Carlo simulation. However, this method has a rather slow enumeration speed through character-by-character generation. We use the default configuration in [18] (hidden layer = 512, training chunk = 256, layers = 2, model optimizer = Adam, generations = 20, and probability threshold = 10^{-10}).

Another method was proposed in 2018. Hitaj et al. used a generative adversarial network to learn the probability distribution from leaked passwords (denoted as GAN) [15]. The generating network model [45] is challenging for natural language modeling because the text is a discrete sequence. IWGAN (Improved Training of Wasserstein GAN) successfully solved this problem by introducing Wasserstein distance, and it was applied to generate text sequences [46]. In [15], generator and discriminator are both convolutional neural networks. After training, the generator can quickly generate massive candidate passwords in batches. We also use the default configuration as with [15] (number of iterations = 200000, discriminator iterations per generator iteration = 10, and size of the input noise vector (seed) = 128).

Both neural network-based methods need a longer time to train and generate passwords compared with other methods. Our GPU-based environment and configuration take 20 hours for training and three days for generating on average. Since we only consider offline cracking, the attacker's cost depends highly on the training and generating time of each method. From this point of view, these methods' cost ranking is followed as rule-based methods < Markov-based methods < PCFG-based methods < neural network-based methods.

4. Empirical Study of Password-Cracking Methods

In this section, we describe the evaluation settings, including the classification of cracking scenarios, the selection of training and targeting datasets, and experiment environment. Next, we perform the empirical evaluation of cracking methods under two cracking scenarios. Then, we conduct further analysis of cracked passwords in Section 4.2. Finally, we summarize our insights.

4.1. Evaluation Settings

4.1.1. Cracking Scenarios. In this paper, we conduct two classes of empirical evaluation in our focused scenario of offline cracking:

(i) Cracking under extensive-knowledge: the attacker knows the exact distribution of target hashed passwords, which is the same assumption as proposed by [47]. Under extensive-knowledge cracking, the attacker has the option to select a training dataset utilizing known target password distribution, which means the attacker can crack using passwords created on the same website as the target. To simulate this scenario, we use part of one dataset as training passwords and the rest of it as targeting passwords.

(ii) Cracking under limited-knowledge: this scenario mainly emphasizes the regional or language differences in password datasets concluded from Section 3.1.2 and whether each method can generalize across password datasets. Limited-knowledge at this moment means the attacker only knows the name, regional, or language information of the target. The attacker does not know the exact distribution of target hashed passwords and can only use passwords from a different source. This represents one situation that the attacker wants to crack hashed passwords that have not been decrypted as plaintext before on this website. We use one dataset as training passwords and other different datasets as targeting passwords to simulate this scenario.

4.1.2. Training and Targeting Datasets. We evaluate all the cracking methods mentioned above with datasets as detailed in Table 1. In order to guarantee fairness and preclude the impact caused by dataset size differences in the evaluation and quantification, we employ a random sampling and adopt a cross-validation approach of 5-fold. We randomly split each dataset and use any four of them (80%) as training data to train each password-cracking method. We see the remaining one (20%) as targeting data to measure cracking efficiency by calculating the number of passwords matched by candidate passwords generated from corresponding methods. The more, the better.

4.1.3. Experiment Environment. We run all of our experiments on a server with the configuration of Redhat 6.7 with 224 GB RAM, a 3.2 GHz Intel Xeon CPU with 32-core, and NVIDIA Titan XP GPU with 12 GB of global memory. For each scenario, we tend to analyze as many passwords as computationally feasible. Here, due to this paper's primary purpose, we limit each method to generate one billion orders of magnitude of candidate passwords according to the actual condition and time required. Except for GAN, since it generates duplicate passwords, and some small-sized datasets such as Phpbb cannot generate many such passwords. Because of these factors, we only compare methods directly at equivalent numbers.

4.2. Evaluation of Password-Cracking Methods

4.2.1. Cracking under Extensive-Knowledge. This scenario is designed to evaluate how well each method can crack passwords when the training and targeting datasets have the same distribution or come from one website. We randomly choose 80% of one dataset as training passwords and the rest as targeting passwords, i.e., when using 80% of the Rockyou dataset for training, the rest 20% is seen as the target. In this scenario, 7 cracking methods' crackability against 12 password datasets is evaluated. We show the cracking results in Table 2. From Table 2, we have several observations as follows:

(i) First of all, it is evident that there is no single optimal password-cracking method which can surpass others in all aspects. Under our settings, the OMEN-based method has the best performance when cracking Phpbb. The PCFG-based method has the best performance when cracking 163, Rockyou. Best64 has the best performance when cracking Duduniu and 000Webhost. FLA has the best performance when cracking 178, CSDN, Sinaweibo, and Duowan. GAN does not work very well.

(ii) For the Markov-based method, OMEN does not show stability when increasing the order from 4 to 5, which is inconsistent with expected results. When cracking CSDN, Sinaweibo, Duowan, and Phpbb, OMEN works better than OMEN-5 with the same candidate password number. Thus, we believe that the higher order of this method is easily overfitting in the training phase.

(iii) For the two versions of PCFG-based methods, PCFG-4 always performs better than PCFG, which is because PCFG-4 is integrated with the Markov model and has richer structures other than LDS in the grammar during the training phase. However, the gap between these two methods can sometimes be larger or smaller, like when cracking 178, Xato, and Yahoo.

(iv) Two neural network-based methods, FLA and GAN, show quite different results. GAN does not exceed any of one method in these datasets and even worse when cracking Duduniu and 000Webhost. On the contrary, FLA demonstrates surprisingly good results, which works best on both Chinese datasets such as CSDN, Sinaweibo, and Duowan, and English datasets such as Xato, Gmail. We argue that the reason is probably related to the principle under these two methods. GAN bases on the generative model, while FLA bases on the probabilistic model. In this situation, the password-generating phrase is more suitable for the probabilistic model since passwords are usually characteristic short strings as in previous studies.

(v) The rule-based method is unstable, but it has a good effect on average. When cracking Duduniu and 000Webhost, it works much better than other methods.

(vi) Although none of these methods are always the best choice, we find there are certain rules to follow by summarizing the optimal password-cracking

Table 2: Empirical evaluation of cracking under extensive-knowledge.

Training datasets	Best64 (%)	OMEN (%)	OMEN-5 (%)	PCFG (%)	PCFG-4 (%)	FLA (%)	GAN (%)
163	52.92	52.09	46.15	57.91	60.44	56.75	40.34
Duduniu	60.88	43.74	44.12	51.55	52.54	49.12	34.06
178	72.77	63.96	64.28	67.36	69.39	77.49	49.51
CSDN	46.75	41.09	38.81	47.02	47.25	56.80	31.90
Sinaweibo	53.75	56.97	55.94	49.64	58.38	67.10	43.61
Duowan	40.96	49.88	46.35	41.33	49.04	58.55	30.71
Rockyou	67.60	53.36	55.46	69.34	69.99	77.91	32.34
000Webhost	43.37	16.56	17.81	30.26	35.35	16.39	7.94
Xato	61.34	47.90	47.73	60.33	62.35	68.63	32.92
Gmail	52.32	45.74	50.05	54.72	57.27	63.22	28.83
Yahoo	38.66	42.46	43.46	48.27	49.61	50.27	26.12
Phpbb	40.93	54.49	44.92	43.21	53.11	51.42	37.80

[1]Each value in this table represents the fraction of passwords been cracked in a dataset (e.g., 52.93% means that 52.92 percent passwords of 163 targeting dataset have been cracked by 163-trained Best64).

method on each password dataset. For example, datasets, such as 178, CSDN, Sinaweibo, and Duowan, have a lower resistance to offline attacks against Markov-chain based methods, including OMEN and FLA, compared with others. These methods generate candidate passwords by predicting the probability of the next characters after a given substring. Also, these password datasets are mostly from Chinese websites. They have apparent, superficial structure distribution characteristics with more passwords composed with a single character and usually have much shorter passwords with a length of less than ten based on Section 3.1.2. Then, for datasets such s 163, Duduniu, Xato, and Gmail, the PCFG-based method has a higher crackability. From Section 3.1.2, we can see that these datasets always have abundant or unique structures, and their average password length is longer than others. Most of them are from English websites. Other than that, Best64 can crack more passwords when the size of training datasets increases. GAN works slightly better when the training datasets have a smaller size.

Therefore, we conclude that no cracking method may outperform others from all aspects in this scenario and is determined by multiple factors, including the underlying model principle and dataset attributes such as length and structure characteristics. In the next section, we will elaborate on further evaluation by studying cracked passwords in each targeting dataset to interpret and verify how these particular characteristic factors affect the cracking effect. Moreover, we will explain how attackers can select a more effective password-cracking method based on the analysis.

4.2.2. Cracking under Limited-Knowledge. In this scenario, we focus on regional or language differences in password datasets. To solve the problem that given one only has Chinese or English website passwords, how well can the attacker crack other websites hashed passwords, and whether the attacker can take advantage of these regional characteristics to make cracking more effective, we carry out cross simulation of passwords created by different language users to observe the effect of regional difference, i.e., when using Rockyou as a training dataset, the target is from the other such as Chinese dataset CSDN or English dataset Phpbb. Due to space limitations, we only show partial results in Table 3, respectively. From Table 3, we have several observations as follows:

(i) First of all, we find that it always achieves better cracking performance when training and targeting datasets are from the same language background. For example, CSDN-trained methods are better at cracking Chinese password datasets such as 163 and 178, while Rockyou-trained methods are always better at cracking English password datasets. This means regional difference does positively affect offline cracking.

(ii) Results show that no single method is optimal in this scenario either under our settings. However, when the training datasets are created under the English website and targeting ones are created under Chinese. Markov-Chain based methods include OMEN and FLA have a far better performance than others. While training datasets are created under the Chinese website and targeting ones are created under English, PCFG-based methods always work better than others. Besides, when the training and targeting datasets are both created under the English website, PCFG-based methods always work better than OMEN. On the contrary, when they are all created under a Chinese website, Markov based methods have a better performance over PCFG. Other than that, Best64 shows relatively better performance than cracking under extensive-knowledge. However, GAN is not satisfactory in this scenario either.

Thus, it is essential to choose a proper training dataset for cracking because user-chosen passwords are more likely to follow their language patterns. If the language and regional information of targeting datasets is available as an auxiliary, an appropriate method could be chosen to achieve a better effect in this scenario. In the next section, we will perform further evaluation by analyzing the distribution of cracked passwords in each targeting dataset to show how attackers can utilize these regional characteristics to select a method that makes cracking more effective.

Table 3: Empirical evaluation of cracking under limited-knowledge.

Targeting datasets	Best64 (%)	OMEN (%)	PCFG (%)	PCFG-4 (%)	FLA (%)	GAN (%)
			Trained on Rockyou			
163	34.68	35.86	30.01	36.27	47.29	18.15
178	39.82	37.76	38.33	42.08	53.06	19.36
Duowan	25.29	27.80	21.62	27.23	40.92	9.87
CSDN	29.92	30.17	28.70	31.98	36.97	13.28
000Webhost	21.53	14.05	29.21	24.23	28.53	5.44
Gmail	54.09	41.39	56.76	55.88	64.65	23.11
Phpbb	58.46	44.16	59.48	58.95	68.24	27.34
Yahoo	54.38	39.52	59.23	55.66	63.75	21.63
			Trained on Xato			
163	31.01	33.43	23.86	34.03	45.48	22.48
178	36.79	39.65	34.53	41.83	51.21	23.95
Duowan	21.22	26.12	16.73	25.52	38.42	12.28
CSDN	28.10	29.50	26.21	30.99	35.40	15.11
000Webhost	19.07	13.97	27.07	23.46	26.44	5.18
Gmail	49.36	39.38	51.16	52.45	60.21	23.53
Phpbb	55.84	44.53	55.78	57.54	65.39	29.17
Rockyou	55.69	45.38	57.36	60.06	67.61	30.39
			Trained on Yahoo			
163	13.54	28.27	13.89	20.62	29.06	13.84
178	19.78	33.75	25.02	28.99	32.07	17.94
Duowan	8.34	20.08	10.30	14.64	18.99	6.52
CSDN	17.25	27.72	18.74	23.77	24.63	6.61
000Webhost	10.40	14.79	20.60	18.85	15.77	8.32
Gmail	31.53	41.49	37.66	42.61	44.86	25.80
Phpbb	37.59	44.73	41.07	46.23	49.30	30.91
Rockyou	37.01	48.60	43.73	50.74	52.84	33.37
			Trained on Duduniu			
000Webhost	16.69	8.56	20.32	17.50	16.34	4.85
Gmail	44.71	19.90	43.34	40.96	39.98	15.90
Phpbb	51.09	19.90	47.75	45.44	42.34	17.85
Rockyou	52.47	21.62	50.87	48.66	48.75	19.32
163	49.31	48.88	43.20	46.86	54.32	41.89
178	57.30	57.37	54.18	56.77	56.63	48.37
Duowan	43.61	48.02	39.36	43.93	52.29	36.52
CSDN	40.37	42.86	38.71	39.34	46.49	30.86
			Trained on CSDN			
000Webhost	10.32	5.24	14.29	12.34	12.27	3.47
Gmail	27.10	10.91	27.37	25.93	29.84	9.53
Phpbb	32.36	9.48	30.52	28.92	32.23	8.97
Rockyou	30.75	8.76	30.14	27.51	31.25	9.43
163	38.78	30.64	37.01	37.10	48.89	32.79
178	46.46	37.94	47.34	45.51	54.67	38.25
Duowan	33.29	29.01	34.45	34.15	46.58	27.23
Sinaweibo	43.46	30.81	40.74	40.28	50.54	33.33
			Trained on Sinaweibo			
000Webhost	14.41	10.18	22.55	20.17	18.06	4.48
Gmail	41.02	31.75	42.99	45.54	48.35	17.36
Phpbb	48.00	35.36	47.47	49.82	53.63	19.62
Rockyou	48.68	38.02	50.49	54.23	57.26	21.48
163	39.76	51.86	35.91	45.97	55.90	40.55
178	45.81	58.42	45.89	53.96	62.52	44.28
Duowan	32.09	48.32	31.27	40.38	52.12	34.85
CSDN	35.09	42.55	35.68	39.46	46.49	29.01

[1] Each value in this table represents the fraction of passwords been cracked in a dataset (e.g., 34.68% indicates that 34.68 percent passwords of 163 targeting dataset have been cracked by Rockyou-trained Best64).

4.3. Further Evaluation of Cracked Passwords. Based on the empirical evaluation above, we conduct further evaluation by examining the cracked passwords of each targeting dataset in each cracking scenario. It is worth noting that each generated candidate dataset matches a very different subset of the target when evaluated in Section 4.2, which means that we gain totally different cracked password datasets. Thus, we conduct further analysis about cracking efficiency in each scenario using a new measurement, which calculates the percentage of each type of password in the cracked dataset accounts for the corresponding subset of passwords in the targeting dataset. For example, there are d passwords of type I in the target. Candidate passwords generated by method A can match $m/d\%$ of type I passwords in the target dataset, while method B can only match $n/d\%$ ($n < m$), which shows that method A is better at cracking passwords of type I. This means each method is better at capturing some distribution features in the training datasets, which results in different cracking capabilities. In this section, our evaluation procedure is followed by classifying the original targeting datasets and each cracked password dataset according to the factors mentioned above. Then, we calculate the percentage of each type of password in the cracked dataset accounts for the corresponding subset of passwords in the targeting dataset.

By analyzing the percentage of various types of passwords cracked by each method, we can empirically deduce the specific types of passwords that each approach is better at cracking. Thus, we can elaborate on how particular characteristics, including length distribution, structure composition, regional differences of the training and targeting datasets, and other dataset attributes, affect the efficiency of cracking methods. This enables us to understand critical factors that affect cracking efficiency and shed light on how to increase password coverage by evaluating password datasets' characteristics.

4.3.1. Cracked Passwords under Extensive-Knowledge

(1) Length Based Evaluation. Results are shown in Figure 4 [1–6], and we have the following observations.

Generally speaking, a longer password is much more difficult to crack, and cracking efficiency decreased as length increased. Other than that, we have some observations:

(i) The PCFG-based method is better at cracking longer passwords (length ≥ 9) than OMEN. From Figure 4, PCFG always cracks a greater percentage of passwords longer than 9, while OMEN can crack more of shorter passwords. This is because PCFG generates candidate passwords based on learning structure distribution. Longer passwords always make up with frequent structures such as LD, LDL, or others, and PCFG prefers generating those structures, which leads to generating large quantities of longer passwords. In comparison, Markov is good at processing sequence through the preceding context and generates candidate passwords based on the Markovian state graph so that shorter passwords would be generated more often.

(ii) FLA can crack much more targeting passwords with lengths longer than ten compared with others. This is because Markov models usually overfit if given too much context, but the neural network typically does not show such quality. We use ten context characters as proven to be successful at guessing in [18], so the context information learned from the training set is more comprehensive than others. Increasing the number of context characters also increases training time, but this could increase accuracy potentially. Thus, FLA performs better on cracking passwords with a longer length.

(iii) GAN shows bad results when cracking longer passwords and only can crack passwords with length less than 12 in some datasets. Our understanding is that GAN works more suitable for training high-frequency length passwords due to IWGAN is a generative model, and it randomly generates candidate passwords as close to the training dataset as possible in batches.

(iv) There is no obvious pattern in the rule-based method. We think it is mainly because rule-based methods generate candidate passwords by traversing all rules. When the rule is more in line with the target, the crackability would be better. We can see that Best64 performs better on cracking passwords with more than 12 letters. The reason is that the Best64 ruleset has a high proportion of adding operations.

(2) Structure-Based Evaluation. Note that we only consider ten popular password structures based on Section 3.1.2. Results are shown in Figure 4 [7–12], and we have the following observations.

Generally speaking, cracked datasets are composed of a large portion of passwords with structures of LD, L, and D. It is easy to explain that these are the top 3 popular structures, which means more training passwords with similar composition are available. Also, these compositions are quite simple, so that the searching space is relatively small. Moreover, most cracked Chinese passwords have a structure D, while most of the cracked English passwords are with structure LD. Specifically,

(i) PCFG is more powerful in cracking passwords composed of more than one type of character such as LD or DL and works much better than any other method when cracking passwords with LD. Even for the hardest password datasets, it can crack 40% of passwords with LD structure in the targeting dataset. However, when it comes to passwords composed of only one type of symbol, the performance is not very well. This can be explained that richness of structure is an important factor that affects the performance of PCFG, and it can learn more structural information than others so that passwords with more complex structures usually have a higher probability of being cracked. Moreover, this verifies the finding that PCFG has the best

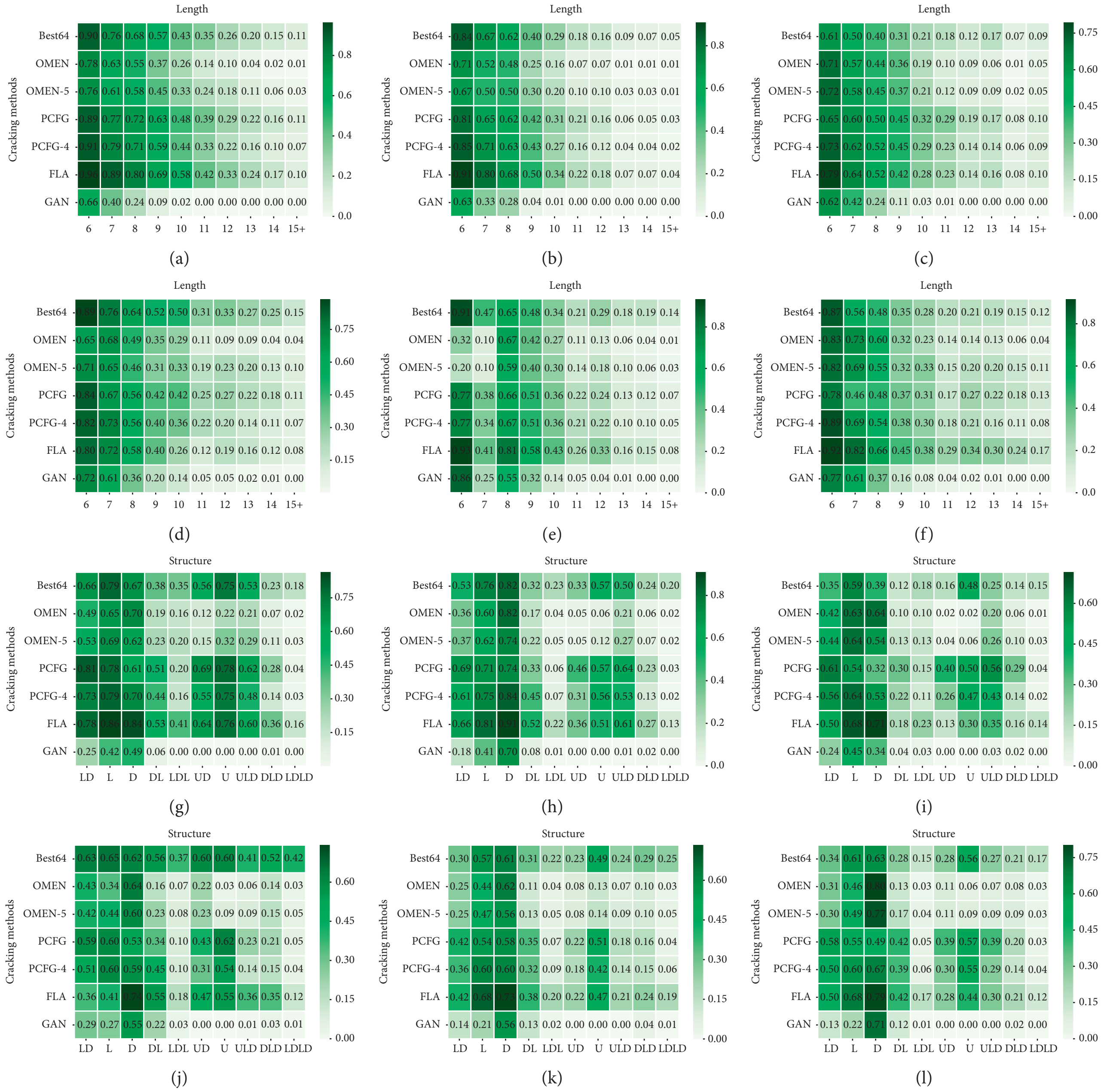

Figure 4: Further evaluation of cracking under extensive-knowledge. Each value in this figure represents the fraction of each length or structure type of passwords in the cracked dataset accounts for the corresponding subset of passwords in the targeting dataset (e.g., in [1], 0.90 indicates that 90 percent passwords with length of 6 in the Rockyou targeting dataset can be cracked by the Rockyou-trained Best64 method). (a) Rockyou-length based evaluation. (b) Xato-length based evaluation. (c) Yahoo-length based evaluation. (d) Duduniu-length based evaluation. (e) CSDN-length based evaluation. (f) Sinaweibo-length based evaluation. (g) Rockyou-structure based evaluation. (h) Xato-structure based evaluation. (i) Yahoo-structure based evaluation. (j) Duduniu-structure based evaluation. (k) CSDN-structure based evaluation. (l) Sinaweibo-structure based evaluation.

performance with most English datasets with more LD structure and few pure digital passwords.

(ii) Markov based methods work best when cracking passwords composed of one type of character. They can crack the most considerable portion of passwords with the structure of D in the target compared with any other methods, especially. However, it does not perform well when passwords contain upper letters. The reason is that most users prefer to choose consecutive characters of the same type as a unit for memorability according to the habit of creating passwords. There is a higher probability that the next character is the same type as the previous one, which results in a small search space. Markov based methods use the idea of Markov-Chain theory so that it can learn more context information of consecutive characters. This also verifies that OMEN has better performance against

Chinese datasets, which consisted of a larger portion D structure.

(iii) FLA works well on all common structures, including both simple and more complex ones. It appears that FLA combines the advantages of both Markov and structure-based methods, which can learn the structural features of passwords from higher dimensions and can construct a large quantity of diverse and novel candidate passwords efficiently.

(iv) The Rule-based method can gain a better crackability for some structures, but there is no obvious pattern. Maybe because these rules have been optimized over the years on password datasets, including Rockyou, it can crack more passwords matching with available rules. In that case, the rule-based method can be complementary to other methods.

(v) Surprisingly, GAN cannot crack complex structure passwords as many as others. We argue that GAN does not gain any prior knowledge of passwords, so it is more likely to generate high-frequency structures as in the training datasets.

In summary, each cracking method has different preferences for learning ability in terms of length and structure distribution. This provides a reasonable strategy for the attacker: the analysis results of training and targeting datasets can be used as the basis for selecting a more effective password-cracking method.

4.3.2. Cracked Passwords under Limited-Knowledge

(1) Length-Based Evaluation. We illustrate the results in Figures 5 and 6 [1–6] and have the following observations.

It is obvious that longer passwords are much more challenging to crack either, and the performance of all cracking methods deteriorates as the length increases. Other than that, Figure 5 shows that the distribution of password length in training, targeting, and cracked datasets is always inconsistent in this scenario. This is because password datasets obtained by the attacker are inconsistent with the distribution of the target. Thus, the cracking method cannot generate passwords similar to the target's distribution by learning the distribution of the training dataset. We observe that when the training and targeting datasets are created under the same language website, which means the length distribution of training and targeting dataset is similar, cracked datasets with length distribution by each method are basically consistent with those of training datasets; while when they are created under different language websites, the length distribution of the cracked datasets is consistent with the target somehow.

(2) Structure-Based Evaluation. Results in Figures 5 and 6 [7–12] show that cracked password datasets are composed of a large portion of passwords with structures of LD, L, and D. Most cracked Chinese passwords have a structure D while most of the cracked English passwords have a structure LD. Besides, each method's characteristics are the same as cracking under extensive-knowledge, including Markov-based methods work best when cracking passwords composed of one type of character, especially with the structure of D and so on in this scenario.

Besides, from Figure 5, the composition of password structure in training, targeting, and cracked datasets is inconsistent sometimes:

(i) When the training and targeting datasets are created under the same language website, there is a certain degree of similar distribution between them. The cracked datasets' structure distribution is basically consistent with those in the training dataset. In order to achieve better tracking performance, we tend to select a method that can capture the distribution characteristics in the training passwords better than others. Thus, the candidate passwords could be more matching with the target can be obtained. So, we assert that when the attacker knows language background of their targeted users and obtains passwords leaked from the same language website, one can choose the right method according to the training dataset's distribution using the conclusion in Section 4.2.1. Maybe it cannot be generalized in some cases, using the method based on this evaluation can always obtain better crackability than blind selection.

(ii) When the training and targeting datasets are created under different language websites, which means the structure distribution of training datasets is totally inconsistent with those in the target. However, the structure distribution of cracked passwords is consistent with the target somehow. We argue that the main reason for this difference is that there is a huge gap between training and targeting datasets in terms of structural distribution, and methods can only generate candidate passwords by learning the structure distribution from a training dataset that cannot capture the targeting dataset distribution characteristics. It is reasonable to choose a cracking method that is more suitable for the distribution characteristics of targeting datasets in order to crack more passwords. In this way, passwords with more common structures can be cracked as many as possible. This verifies the finding in Section 4.2.2. So, we assert that when an attacker obtains passwords leaked in a different region from the target, one can choose the right method according to the target using the conclusion in Section 4.2.1 in this case.

4.4. Insights. Our evaluation results show how comparative analyzes uncover each approach's relative superiority under these two offline cracking scenarios. Upon further examination, many cracking behaviors make sense. We suggest exploring the password creation strategy from the perspective of length and structure characteristics, mainly prioritizing the structure distribution so that one can select a more proper method. To be specific,

(i) When *cracking under extensive-knowledge*, if the majority of passwords in the targeting dataset are

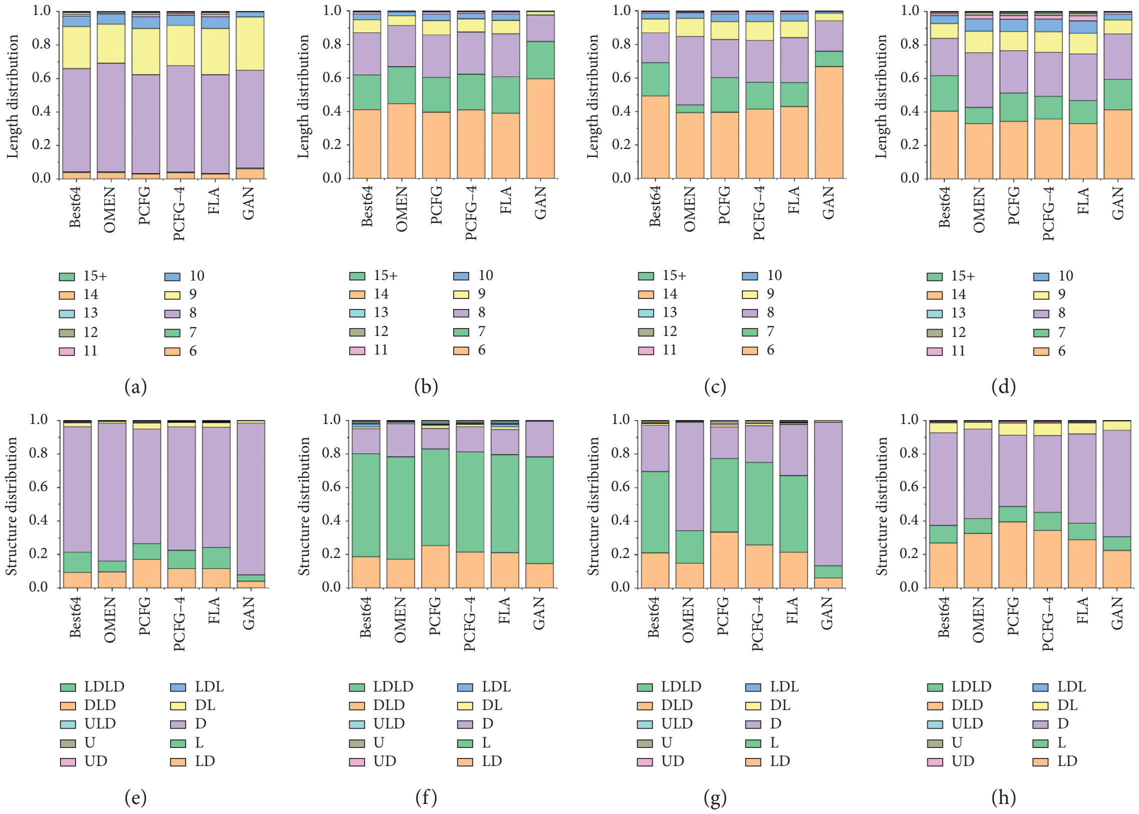

Figure 5: Further evaluation of cracking under limited-knowledge I. Each figure shows the distribution of passwords cracked by the corresponding method in this scenario (e.g., the first one shows length distribution of the CSDN targeting dataset cracked by the Rockyou-trained variety method). (a) Trained on Rockyou and tested on CSDN-length based evaluation. (b) Trained on Rockyou and tested on Phpbb-length based evaluation. (c) Trained on CSDN and tested on Rockyou-length based evaluation. (d) Trained on CSDN and tested on 178-length based evaluation. (e) Trained on Rockyou and tested on CSDN-length based evaluation. (f) Trained on Rockyou and tested on Phpbb-length based evaluation. (g) Trained on CSDN and tested on Rockyou-length based evaluation. (h) Trained on CSDN and tested on 178-length based evaluation.

composed of one type of characters like most Chinese password datasets, especially with the structure of D, the Markov-based method, OMEN and FLA, is a better choice. While most passwords consist of more than one type of character like most English datasets, particularly with the structure of LD, PCFG works much better. Although the two neural network-based methods have similar more considerable cracking costs, FLA demonstrates a surprisingly good effect on both Chinese and English datasets. In contrast, GAN does not perform well compared to others. The rule-based method is not stable, but it can gain a better crackability when the datasets have a larger size or more complex structures. Besides, OMEN is better at cracking shorter passwords (length < 9), and PCFG is better at cracking longer passwords. Also, except for higher computational cost, FLA could crack more passwords with a longer length of more than 11.

(ii) When *cracking under limited-knowledge*, language background or region information plays a significant role, which does affect the choice of cracking methods. When the training dataset is created under the same language website as the targeting dataset, attackers should choose the cracking method based on the training dataset's distribution. It will always gain much better performance than others. Otherwise, one should choose the right method according to the targeting dataset.

5. Discussion

5.1. Limitations. We discuss some limitations as follows. First, the 12 datasets collected in this paper were all leaked from Chinese and English websites. Note that Chinese and English netizens are the most considerable fraction of the world's Internet population [48]. It is acceptable for these datasets on behalf of current password users' practice. Analysis of passwords in other or less widespread languages will be studied in the following work. Simultaneously, there are likely to be contamination issues because datasets are directly accessed from the Internet. Other than unreasonable passwords that have been filtered out, we regard this as the inevitable uncertainty in password creation since absolute randomness in password creation cannot be generalized. Second, we argue that 10^9 is an appropriate candidate password number for offline cracking evaluation. Although

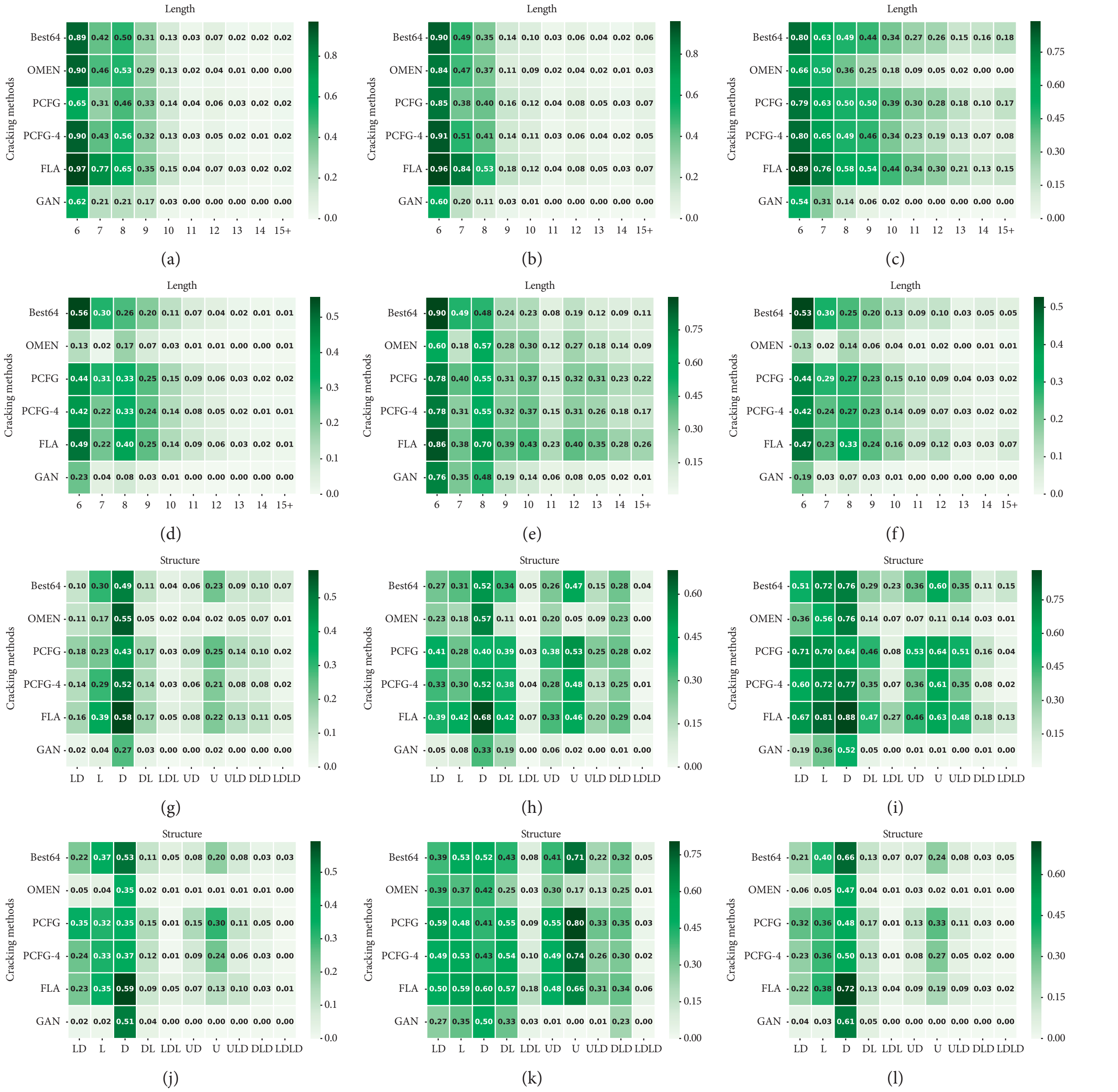

Figure 6: Further evaluation of cracking under limited-knowledge II. Each value in this figure represents the fraction of each length or structure type of passwords in the cracked dataset accounts for the corresponding subset of passwords in the targeting dataset in this scenario (e.g., in [1], 0.89 indicates that 89 percent passwords with length of 6 in the CSDN targeting dataset can be cracked by the Rockyou-trained Best64 method). (a) Trained on Rockyou and tested on CSDN-length based evaluation. (b) Trained on Rockyou and tested on 178-length based evaluation. (c) Trained on Rockyou and tested on Phpbb-length based evaluation. (d) Trained on CSDN and tested on Rockyou-length based evaluation. (e) Trained on CSDN and tested on 178-length based evaluation. (f) Trained on CSDN and tested on Phpbb-length based evaluation. (g) Trained on Rockyou and tested on CSDN-structure based evaluation. (h) Trained on Rockyou and tested on 178-structure based evaluation. (i) Trained on Rockyou and Tested on Phpbb-structure based evaluation. (j) Trained on CSDN and tested on Rockyou-structure based evaluation. (k) Trained on CSDN and tested on 178-structure based evaluation. (l) Trained on CSDN and tested on Phpbb-structure based evaluation.

it is feasible to generate more passwords directly or even exhaust the entire password space, we focus on implementing cracking methods in a uniform environment to make the comparison fair and reduce possible bias. Besides, it takes significantly more time for some methods to generate a large number of passwords. For instance, it takes the original PCFG or FLA several days to generate one billion passwords. Furthermore, it takes about two weeks with 4 TiTan XP to try 10^9 guesses against 8 million MD5 string with salt, and most of the websites use more complicated hash schemes such as SHA256 or Scrypt [49]. So, we think it is also beyond concern about cracking efficiency [41]. We consider these computational limitations are essential, and attackers should pay attention to it in practice. Third, since there are no publicly available expired/reused leaked password datasets and using passwords with personal information of any users may raise ethical concerns, we only consider methods that do not involve these kinds of datasets.

5.2. Future Work. First, we will evaluate more password datasets versus new cracking algorithms as a supplement in the following work. Second, we only find out some dominant characteristics in passwords, and there is plenty of other factors that can be explored in the future to improve cracking results in practice further. We will extent experiments to get more information on passwords. Third, we will study whether the combination method can effectively improve cracking efficiency.

6. Conclusion

In this paper, we conduct a large-scale empirical study on password-cracking methods proposed by the academic community since 2005, leveraging 220 million plaintext passwords leaked from 12 popular websites during the past decade under two offline cracking scenarios. Studying and summarizing state-of-the-art cracking methods can help to design more secure authentication schemes that can resist such attacks. Subsequently, we present further evaluation by analyzing the set of cracked passwords in each targeting dataset. Some suggestions are given on how to choose a more effective password-cracking method to achieve the goal of accurate evaluation when conducting offline cracking under these two scenarios. Based on our evaluation results, one can gain a deeper understanding of selecting a cracking method to make a fair and impartial evaluation of password-based authentication systems resistance against the most robust offline crack.

Data Availability

All the password datasets used to support the findings of this study are publicly available for downloading.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (Nos. 61632020, U1936209, 62002353, and U1836207) and Beijing Natural Science Foundation (No. 4192067).

References

[1] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "The quest to replace passwords: a framework for comparative evaluation of web authentication schemes," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy (S&P)*, pp. 553–567, IEEE, San Francisco, CA, USA, May 2012.

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

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

[4] D. Florêncio, C. Herley, and P. C. Van Oorschot, "An Administrator's Guide to Internet Password Research," in *Proceedings of the 28th Large Installation System Administration Conference (LISA14)*, pp. 44–61, Seattle, WA, USA, November 2014.

[5] R. Martin, "Amid widespread data breaches in China," December 2011, http://www.techinasia.com/alipay-hack.

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

[7] D. Goodin, "Personal data is exposed as a result of a five-month-old hack on 000webhost," October 2015, http://t.cn/R4tKrEU.

[8] R. Morris and K. Thompson, "Password security," *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[9] P. G. Kelley, S. Komanduri, M. L. Mazurek et al., "Guess again (and again and again): measuring password strength by simulating password-cracking algorithms," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP)*, pp. 523–537, IEEE, San Francisco, CA, USA, May 2012.

[10] B. Ur, S. M. Segreti, L. Bauer et al., "Measuring real-world accuracies and biases in modeling password guessability," in *Proceedings of the USENIX Security Symposium*, pp. 463–481, Washington, DC, USA, August 2015.

[11] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[12] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password Cracking Using Probabilistic Context-free Grammars," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, pp. 391–405, IEEE, Berkeley, CA, USA, May 2009.

[13] J. Ma, W. Yang, M. Luo, and N. Li, "A Study of Probabilistic Password Models," in *Proceedings of the 2014 IEEE Symposium onSecurity and Privacy (SP)*, pp. 689–704, IEEE, San Jose, CA, USA, May 2014.

[14] M. Dürmuth, F. Angelstorf, C. Castelluccia, D. Perito, and A. Chaabane, "Omen: faster password guessing using an ordered markov enumerator," in *Proceedings of the International Symposium on Engineering Secure Software and Systems*, pp. 119–132, Springer, Milan, Italy, March 2015.

[15] B. Hitaj, P. Gasti, G. Ateniese, F. Perez-Cruz, and " Passgan, "A deep learning approach for password guessing," in *Proceedings of the International Conference on Applied Cryptography and Network Security*, pp. 217–237, Springer, Bogotá, Colombia, June 2019.

[16] M. Dell'Amico, P. Michiardi, and Y. Roudier, "Password strength: an empirical analysis," in *Proceedings of the 2010 Proceedings IEEE INFOCOM*, pp. 1–9, IEEE, San Diego, CA, USA, August 2010.

[17] M. L. Mazurek, S. Komanduri, T. Vidas et al., "Measuring password guessability for an entire university," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, pp. 173–186, ACM, Berlin, Germany, November 2013.

[18] W. Melicher, B. Ur, S. M. Segreti et al., "Fast, lean, and accurate: modeling password guessability using neural

networks," in *Proceedings of the 25th USENIX Security Symposium (USENIX Security 16)*, pp. 175–191, USENIX Association, Austin, TX, USA, August 2016.

[19] S. Ji, S. Yang, X. Hu, W. Han, Z. Li, and R. Beyah, "Zero-sum password cracking game: a large-scale empirical study on the crackability, correlation, and security of passwords," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, pp. 550–564, 2017.

[20] A. Narayanan and V. Shmatikov, "Fast dictionary attacks on passwords using time-space tradeoff," in *Proceedings of the 12th ACM Conference on Computer and Communications Security*, pp. 364–372, ACM, Alexandria, VA, USA, November 2005.

[21] Z. Li, W. Han, and W. Xu, "A large-scale empirical analysis of Chinese web passwords," in *Proceedings of the 23rd USENIX Security Symposium (USENIX Security 14)*, pp. 559–574, USENIX Association, San Diego, CA, USA, August 2014.

[22] R. Veras, C. Collins, and J. Thorpe, "On semantic patterns of passwords and their security impact," *In NDSS*, 2014.

[23] S. Houshmand, S. Aggarwal, and R. Flood, "Next gen PCFG password cracking," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 8, pp. 1776–1791, 2015.

[24] W. Han, M. Xu, J. Zhang, C. Wang, K. Zhang, and X. S. Wang, "TransPCFG: transferring the grammars from short passwords to guess long passwords effectively," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 451–465, 2020.

[25] J. Steube, "Hashcat v4.0.1-30-ge93fa25+," June 2017, https://github.com/hashcat/hashcat.

[26] S. Designer, "John the ripper password cracker," 2006.

[27] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: understanding passwords of Chinese web users," in *Proceedings of the 28th {USENIX} Security Symposium*, vol. 19, pp. 1537–1555, {USENIX} Security, 2019.

[28] K. Mori, T. Watanabe, Y. Zhou, A. Akiyama Hasegawa, M. Akiyama, and T. Mori, "Comparative analysis of three language spheres: are linguistic and cultural differences reflected in password selection habits?" *IEICE Transactions on Information and Systems*, vol. E103.D, no. 7, pp. 1541–1555, 2020.

[29] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse," *In NDSS*, vol. 14, pp. 23–26, 2014.

[30] W. Han, Z. Li, M. Ni, G. Gu, and W. Xu, "Shadow attacks based on password reuses: a quantitative empirical analysis," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 2, pp. 309–320, 2018.

[31] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: an algorithmic framework and empirical analysis," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 176–186, ACM, Chicago IL USA, October 2010.

[32] S. Ji, S. Yang, A. Das, X. Hu, and R. Beyah, "Password correlation: quantification, evaluation and application," in *Proceedings of the IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pp. 1–9, IEEE, Atlanta, GA, USA, May 2017.

[33] B. Ye, Y. Guo, L. Zhang, and X. Guo, "An empirical study of mnemonic password creation tips," *Computers & Security*, vol. 85, pp. 41–50, 2019.

[34] J. Zeng, J. Duan, and C. Wu, "Empirical study on lexical sentiment in passwords from Chinese websites," *Computers & Security*, vol. 80, pp. 200–210, 2019.

[35] K. S. Walia, S. Shenoy, and Y. Cheng, "An Empirical Analysis on the Usability and Security of Passwords," in *Proceedings of the 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pp. 1–8, IEEE, Las Vegas, NV, USA, August 2020.

[36] M. AlSabah, G. Oligeri, and R. Riley, "Your culture is in your password: an analysis of a demographically-diverse password dataset," *Computers & Security*, vol. 77, pp. 427–441, 2018.

[37] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing metrics for password creation policies by attacking large sets of revealed passwords," in *Proceedings of the 17th ACM Conference on Computer and Communications Security*, pp. 162–175, Chicago, IL, USA, October 2010.

[38] D. L. Wheeler, "zxcvbn: low-budget password strength estimation,"vol. 16, pp. 157–173, in *Proceedings of the 25th {USENIX} Security Symposium*, vol. 16, pp. 157–173, {USENIX} Security, Austin, TX, USA, August 2016.

[39] I. Dropbox, "Dropbox," 2014, http://www.dropbox.com.

[40] X. de Carné de Carnavalet and M. Mannan, "From very weak to very strong: analyzing password-strength meters," in *Proceedings of the Network and Distributed System Security Symposium (NDSS 2014)*, Internet Society, San Diego, California, USA, February 2014.

[41] J. Blocki, B. Harsha, and S. Zhou, "On the economics of offline password cracking," in *Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP)*, pp. 853–871, IEEE, San Francisco, California, USA, May 2018.

[42] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek et al., "How does your password measure up? the effect of strength meters on password creation,"vol. 12, pp. 65–80, in *Proceedings of the Presented as part of the 21st {USENIX}Security Symposium*, vol. 12, pp. 65–80, {USENIX} Security, Bellevue, WA, USA, August 2012.

[43] M. Weir, "pcfg_cracker Version 3.2 C5ca74f," April 2017, https://github.com/lakiw/pcfg_cracker.

[44] ——, "pcfg_cracker v4.1 869fb3d," August 2019, https://github.com/lakiw/pcfg_cracker.

[45] I. Goodfellow, "Nips 2016 tutorial: generative adversarial networks," 2016, https://arxiv.org/abs/1701.00160.

[46] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

[47] J. Bonneau, "The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, pp. 538–552, IEEE, San Francisco, CA, USA, May 2012.

[48] C. I. N. I. Center, "China now has 802 million internet users," July 2018, http://n0.sinaimg.cn/tech/c0a99b19/20180820/CNNIC42.pdf.

[49] J. Blocki, A. Datta, and " Cash, "A cost asymmetric secure hash algorithm for optimal password protection," in *Proceedings of the 2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, pp. 371–386, IEEE, Lisbon, Portugal, June-July 2016.