Zero-Sum Password Cracking Game: A Large-Scale Empirical Study on the Crackability, Correlation, and Security of Passwords

Shouling Ji, *Student Member, IEEE,* Shukun Yang, *Student Member, IEEE,* Xin Hu,Weili Han, *Member, IEEE,* Zhigong Li, and Raheem Beyah, *Senior Member, IEEE*

Abstract—In this paper, we conduct a large-scale study on the crackability, correlation, and security of ~ 145 million real world passwords, which were leaked from several popular Internet services and applications. To the best of our knowledge, this is the largest empirical study that has been conducted. Specifically, we first evaluate the crackability of ~ 145 million real world passwords against 6+ state-of-the-art password cracking algorithms in multiple scenarios. Second, we examine the effectiveness and soundness of popular commercial password strength meters (e.g., Google, QQ) and the security impacts of username/email leakage on passwords. Finally, we discuss the implications of our results, analysis, and findings, which are expected to help both password users and system administrators to gain a deeper understanding of the vulnerability of real passwords against state-of-the-art password cracking algorithms, as well as to shed light on future password security research topics.

Index Terms—Passwords, evaluation, crackability, classification, correlation, password meter, password strength.

1 Introduction

Password-based authentication is the most widely used user authentication method in modern computer systems [23]. Recently, password security has drawn increasing attention from the research community [1][2][5][12][22]. This is probably because of several serious password leakage incidents, (e.g., `CSDN` *password leakage incident* [26], `Yahoo!` *password leakage incident* [27]), and thus people care more about their password security. However, to help both password users and system administrators gain a deeper understanding of the vulnerability of current password systems as well as the threat of modern password cracking algorithms, several open problems in the password security research area still need to be studied. Specifically, (i) *What is the vulnerability of current password systems against state-of-the-art password cracking algorithms, e.g., semantics based password cracking algorithms [12]?* (ii) *What is the effectiveness and soundness of current popular commercial password meters (e.g., Google, QQ password) on helping users secure their passwords?* (iii) *What are the impacts of usernames and emails leakage on passwords' security?* (iv) *What is the correlation among different password systems/datasets?*

Although there are several empirical studies [1]-[9] on password security, unfortunately, none of them comprehensively addressed the open problems mentioned above due to one or several limitations, e.g., emerging password cracking algorithms are not evaluated; only a small password corpus is employed; the impacts of password meters, usernames and/or emails are not considered.

Aiming at addressing the above four open problems, and helping both password users and system administrators update their understanding of the vulnerability of current password systems and the threat of modern password cracking algorithms, we conduct a large-scale empirical study on the crackability, correlation, and security of 15 real world password datasets (~ 145 million passwords) which covers various popular Internet services/applications (see Table 3). Particularly, our contributions are summarized as follows.

(i) We evaluate and analyze the crackability of 15 large-scale real world password datasets (~ 145 million passwords) against 6+ state-of-the-art password cracking algorithms in multiple scenarios, including the *training-free cracking*, *intra-site cracking*, and *cross-site cracking*. We also make several interesting observations (e.g., the *overfitting phenomena* of Markov model based cracking algorithms) and remark on the advantages/disadvantages of the examined password cracking algorithms. Besides traditional password crackability evaluation, we go further by conducting in-depth classification based crackability analysis, which enables password users and administrators to understand the length, structure, and composition characteristics of insecure (or easily-crackable) passwords.

(ii) We evaluate the effectiveness and soundness of commercial password strength meters of popular sites, e.g., Google, Twitter, QQ, 12306.cn. Based on our results, it is evident that some password meters are not currently guiding users to choose secure passwords. Sometimes, they may even mislead users. On the other hand, proper password meters are useful in helping users choose secure passwords

- *S. Ji, S. Yang, and R. Beyah are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: {sji, syang87}@gatech.edu, rbeyah@ece.gatech.edu*
- *X. Hu is with the IBM T. J. Watson Research Center. E-mail: huxin@us.ibm.com*
- *W. Han and Z. Li are with the Software School, Fudan University. E-mail: wlhan@fudan.edu.cn*

against modern password cracking algorithms.

(iii) We evaluate the security impacts of username/email leakage. According to our results, both usernames and email leakage have surprising impact on password security, which alerts users and system administrators to the fact that besides passwords themselves, usernames, email addresses, and other user profiles also deserve dedicated protection.

(iv) We evaluate the correlation among passwords. We find that user-chosen passwords do exhibit *regional/language* (or, *cultural*) *differences*. This finding has implications on how to select proper training data and how to measure the *mutual information* between two password systems.

2 Related Work

Empirical Studies on Password Security. In [4][5][7], the authors evaluated password use/re-use habits and password policies of a number of websites. In [6], Zhang et al. studied the security of password expiration using 7000+ accounts. In [8], Bonneau conducted a study to estimate the password guessing difficulty. In [9], Mazurek et al. implemented another study to measure the password guessability.

In [2], Ma et al. investigated probabilistic password models. Li et al. studied the differences between passwords from Chinese and English users in [1]. The most related work to this paper is [3], where Dell'Amico et al. conducted an empirical analysis on password strength of 58,800 users. However, only the *dictionary attack*, PCFG [11], and a Markov model based scheme [10] were evaluated in [3]. Recently, Ji et al. developed a uniform and open-source password analysis and research system PARS [21].

Password Cracking. In [10], Narayanan and Shmatikov designed a fast dictionary attack on passwords based on a Markov model, which can generate candidate password guesses with probability above some threshold value. Dürmuth et al. improved Narayanan and Shmatikov's Markov model based password cracking in [16] by designing an *Ordered Markov ENumerator* (OMEN). OMEN can make password guesses in the decreasing order of possibility. Password cracking using *Probabilistic Context Free Grammars* (PCFG) was introduced in [11] by Weir et al. Since the semantic information is not considered in [11], in [12], Veras et al. improved PCFG by developing new password grammars that capture structural, syntactic, and semantic patterns of passwords.

Recently, password crackers/utilities also have made great advances, e.g., JtR-Jumbo [13], JtR-Bleeding jumbo [14], and Hashcat [15]. JtR-Jumbo [13] is the latest official community-enhanced version of the popular password cracking software JtR. It supports multiple different password cracking modes, e.g., *dictionary mode*, *Markov mode*, *incremental mode*, and *single mode*. JtR-Bleeding jumbo [14] is the latest academia enhanced version of JtR [14], which provides more functionality and is more powerful than JtR-Jumbo. Hashcat [15] is also a popular password cracker which supports multiple modes, e.g., *dictionary mode*, *Markov mode*, and *mask attack mode*.

Password Measurement and Meters. In [17], Weir et al. examined metrics for password creation policies by performing password attacks. Taking another direction, in [18], Castelluccia et al. investigated adaptive password strength meters from Markov models. Another work to measure password strength by simulating password cracking algorithms is [19], where Kelley et al. examined the resistance of 12K passwords against Weir's PCFG based and a Markov based password cracking algorithms. Recently, password meters have garnered a lot of attention from the research community. In [20], Ur et al. presented a 2,931-subject study of password creation in the presence of 14 password meters. In [22], Carnavalet and Mannan analyzed password meters of popular websites, e.g., Microsoft, Google.

TABLE 1
Dataset statistics. U = *username*, E = *email*, L = *language*, CH = *Chinese*, EN = *English*, GE = *German*, IC = *Internet-cafe Service*, OD = *Online Dating*, and SN = *Social Netowrks*.

name	size	unique	U	E	L	type	date
17173.com	18.3M	5.2M	yes	yes	CH	game	2011
178.com	9.1M	3.5M	yes	no	CH	game	2011
7k7k	12.9M	3.5M	no	yes	CH	game	2011
CSDN	6.4M	4M	yes	yes	CH	programmer	2011
Duduniu	16.1M	10M	no	yes	CH	IS	2011
eHarmony	1.6M	1.6M	no	no	EN	OD	2012
Gamigo	6.3M	6.3M	no	no	GE	game	2012
Hotmail	8.9K	8.9K	no	no	EN	email	2009
LinkedIn	5.4M	4.9M	no	no	EN	SN	2012
MySpace	49.7K	41.5K	no	yes	EN	SN	2006
phpBB	.2M	.2M	no	no	EN	software	2009
Renren	4.7M	2.8M	no	yes	CH	SN	2011
Rockyou	32.6M	14.3M	no	no	EN	game	2009
Tianya	31M	12.6M	yes	yes	CH	forum	2011
Yahoo!	.4M	.3M	no	yes	EN	Internet	2012

3 Password Datasets and Methodologies

3.1 Password Datasets

In this paper, we evaluate the crackability and security implications of 15 leaked password datasets (145 million real world passwords), which are used in various Internet systems and services (e.g., email, gaming, dating) and from multiple language/national domains. We summarize the datasets in Table 1. All the leaked password datasets are a result of *password leakage incidents* and published by unknown individuals/parties [1][2][12]. Furthermore, the provenances of these password datasets varies. They may have ultimately been made available through *SQL injection attacks*, *phishing campaigns*, etc. [5]. In this paper, we only use these datasets for research purposes.

3.2 Methodologies

In this paper, we study the crackability of 145 million real passwords against 6+ modern password cracking schemes, which can be classified into four categories: *password crackers* [13][14][15], *Probabilistic Context Free Grammar (PCFG) based schemes* [11], *semantic pattern based schemes* [12], and *Markov model based schemes* [13][14][15][16].

Password crackers [13][14][15][1]. The popular password crackers considered are JtR 1.7.9-Jumbo-7 (JtR-J) [13], JtR-

1. We also examined the *paid version* JtR *Pro* [13] in our evaluation. However, based on our experience and results, except for the customer service, both JtR-B and JtR-J, especially JtR-B, outperforms JtR *Pro* with respect to *software functionality*, *utility usability*, and *password cracking performance*. Therefore, we did not include the results of JtR *Pro* in this paper.

Bleeding jumbo (JtR-B) [14], and Hashcat [15]. As we mentioned before, JtR-J is the latest *official community-enhanced* version of JtR and JtR-B is the latest *academia enhanced* version of JtR. For both JtR-J and JtR-B, we evaluate both their *dictionary* mode and their *incremental* mode. Under the dictionary mode, password guesses are generated according to an input dictionary. The incremental mode is the most powerful cracking mode of JtR. Under the incremental mode, the entire password space is searched by *intelligent brute force*, where the *statistical character frequencies* are considered. For Hashcat, we evaluate its *dictionary* and *mask attack* modes. The dictionary mode of Hashcat is similar to that of JtR. The mask attack mode of Hashcat is an improved brute force attack, where the patterns of human generated passwords will be considered. When evaluating the dictionary mode of JtR-J, JtR-B, and Hashcat, we use a combined dictionary consisting of *dic-0294* (English word list) [11], *Pinyin* (Chinese word list) [1], *paid JtR dictionary* (includes word lists for 20+ human languages and lists of common passwords) [13], and *keyboard_dic* (keyboard shortcuts) [24].

PCFG schemes [11]. The idea of using PCFG to crack passwords was introduced by Weir. et al. in [11]. Here, each grammar can be viewed as a password structure, e.g., if "D" denotes *digits*, "L" denotes *lower case letters*, "U" denotes *upper case letters*, and "S" denotes *special characters*, password "123456" has a structure of D6, "yo_pendejo_4" has a structure of L2S1L7S1D1, and "#myNAME?66" has a structure of S1L2U4S1D2. The main idea is to create a PCFG based on a password training set. Then, the PCFG is ordered by the probability (frequency) of appearance of each grammar (password structure) from high to low. Finally, the PCFG is used to generate word-mangling rules according to the probability order, followed by password guesses. Note that, during the password guess generation process, the password structure with a high probability will be used first since it is more likely that it will generate a correct password guess. Since the evaluated password datasets cover English, Chinese, and German, the input dictionary for PCFG based password cracking scheme [11] consists of *dic-0294* (English word list, used in [11]), *JtR paid English word list* [13], *Pinyin* (Chinese word list, used in [1]), *JtR paid German word list* [13], *JtR paid password list* [13], and *keyboard_dic* [24].

When using the public version of PCFG [11] to crack large-scale passwords, it has been reported several times that a *segmentation error* (a memory bug) appears [16] (we also found this error in our evaluation). Therefore, while working on this study, we fixed the memory bug of the public PCFG cracker by implementing a *multilevel priority queue based PCFG cracker*. For convenience, we still refer to the improved PCFG cracker as PCFG in the rest of this paper.

Semantic pattern based schemes [12]. Since the PCFG based scheme [11] does not take into account the letter part in a password's structure, recently, Veras et al. proposed to redesign the PCFG by considering both syntactic and semantic patterns of passwords in [12]. Specifically, they first designed a scheme to segment, classify, and generalize semantic categories from training passwords by leveraging Natural Language Processing (NLP) algorithms. Subsequently, based on PCFG, they develop a grammar that captures structural, syntactic, and semantic patterns of passwords. Finally, password guesses will be generated in terms of the new developed grammars.

Markov model based schemes [13][14][15][16]. In [10], Narayanan and Shmatikov proposed a Markov model based scheme to conduct fast dictionary attacks on passwords. The idea is to build a Markov model based on training passwords and then use this model to generate new guesses. A limitation of the Markov model in [10] is that it can only generate passwords whose probability is above some threshold value while not necessarily following a probability decreasing order, which is different from PCFG algorithms [11][12].

Following [10], several improved Markov model based password cracking schemes have been developed and implemented: *JtR 1.7.9-Jumbo-7 Markov mode* (JtR-J-M) [13], *JtR-Bleeding jumbo Markov mode* (JtR-B-M) [14], *Hashcat Markov mode* (Hashcat-M) [15], and *Ordered Markov ENumerator* (OMEN) [16]. JtR 1.7.9-Jumbo-7 is the latest *official community-enhanced* version of JtR with many functions and utilities, and it has a Markov password cracking mode. JtR-Bleeding jumbo is the latest *academia enhanced* version of JtR released at GitHub, which is more powerful than the official JtR 1.7.9-Jumbo-7. It also supports the Markov password cracking mode. Hashcat is a self-proclaimed CPU-based password recovery/cracking tool. It has a Markov model based password cracking mode. OMEN is a recently proposed Markov model based password cracking algorithm, which extends the idea in [10]. In OMEN, the password guesses will be generated in the decreasing order of likelihood.

4 General Evaluation

Evaluation Setup. We mainly conduct three classes of evaluation: *training-free cracking*, *intra-site training and cracking*, and *cross-site training and cracking*, where intra-site means that for each dataset, we use part of its passwords for training and use the rest for testing, and cross-site means that we use specific datasets for training and use the other datasets for testing.

When we conduct intra-site training and cracking, we randomly select $10\% - 50\%$ of the passwords from each dataset for training, and use the rest of the passwords for testing. For cross-site training and cracking, we use `Tianya`, `Rockyou`, and `Tianya+Rockyou` for training respectively, and use all the other datasets for testing. For the cracking schemes that do not have the training phase, we conduct password cracking directly using the testing dataset.

Training-free Password Cracking. In this part of the evaluation, we examine the crackability of the 15 datasets against password cracking algorithms without data training. Specifically, the evaluated algorithms are JtR-J *dictionary mode* (JtR-J-Dic), JtR-B *dictionary mode* (JtR-B-Dic), Hashcat *dictionary mode* (Hashcat-Dic), JtR-J *incremental mode* (JtR-J-Inc), JtR-B *incremental model* (JtR-B-Inc), and Hashcat *mask attack mode* (Hashcat-Mask). Since JtR-J-Dic, JtR-B-Dic, and Hashcat-Dic have the same performance in our evaluation, we use "Dictionary" to represent their results. Furthermore, for JtR-J-Dic, JtR-B-Dic, and Hashcat-Dic, the input dictionary is specified in Section 3.2, and for JtR-J-Inc, JtR-B-Inc,

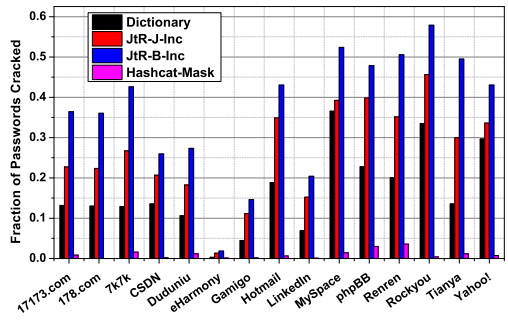


Fig. 1. Password resistance vs cracking algorithms without training data.

and Hashcat-Mask, they all make $\sim 10^{10}$ guesses. We show the results in Fig.1.

From Fig.1, we have the following observations.

(i) Modern training-free cracking algorithms, especially JtR-B-Inc (the most powerful training-free password cracking scheme in our evaluation), are very powerful. For many target datasets, e.g., `Rockyou`, `MySpace`, `Tianya`, even without any priori knowledge, a significant portion of their passwords can be cracked with reasonable computational cost. Furthermore, the crackability of different password systems is very different. Among the 15 studied password datasets, `eHarmony` is the most difficult one to crack. This is because `eHarmony` has more passwords that are longer and more complicated than the other datasets. Meanwhile and surprisingly, `eHarmony` by no means requires a "complicated/strong" password in the registration phase (it only requires 8 or more characters). Since `eHarmony` is an online dating site, people seem to care more about securing their dating information. An in-depth analysis of this phenomenon is presented later.

(ii) According to the input dictionary, Dictionary makes about 134.4 million guesses. Nevertheless, it demonstrates competitive performance and can crack $10\% - 40\%$ of the passwords of most datasets, which indicates that Dictionary is effective with respect to return on guessing. Therefore, Dictionary can be used to quickly break partial passwords of a dataset, which can further serve as auxiliary information for powerful training-based password cracking algorithms.

(iii) JtR-B-Inc and JtR-J-Inc have better performance than Dictionary. This is mainly because they make more guesses. According to our results, Dictionary has better performance if the same number of guesses are made. The reason is evident since the Dictionary based attack tries the most frequently observed passwords (or combinations) in the real world, and a number of people still use the leaked weak passwords even after many password leakage incidents. The most significant advantage of JtR-B-Inc and JtR-J-Inc (as well as Hashcat mask attack mode) over Dictionary (as well as other algorithms, e.g., JtR Markove mode, OMEN, PCFG, semantic based schemes) is that they can intelligently brute force the entire password space. Theoretically, they can crack all the passwords given enough computational power. JtR-B-Inc is also better than JtR-J-Inc for all the considered datasets since it improved the guessing algorithm of JtR-J-Inc by employing better *character frequency statistics*. Furthermore, Hashcat-Mask has the worst performance under the current setting. This is because Hashcat makes guesses following a *length increasing order of candidate passwords*. It easily reaches 10^{10} guesses without generating effective password candidates when considering all the letters, special characters, and digits.

Training-based Intra-site Cracking. In this subsection, we evaluate the training-based intra-site crackability of the 15 password datasets. For each dataset, we randomly select 10%, 30%, and 50% of the passwords as training data, and then try to crack the rest of the dataset based on the trained models. The employed cracking algorithms are PCFG, JtR-J Markov mode with *Markov level* 212 and 215 respectively (*Markov level* is a parameter to control the number of guesses), JtR-B Markov mode with Markov level 212 and 215 respectively, Hashcat Markov mode with *threshold* 4 (a parameter controls the number of guesses), and OMEN. Based on our results, JtR-J and JtR-B have the same performance in the Markov mode. For convenience, we use J212 and J215 to denote JtR-J/JtR-B Markov mode with levels 212 and 215, respectively, and H4 to denote Hashcat Markov mode with threshold 4. We show the cracking results in Table 2.

From Table 2, we have several observations as follows.

(i) When more training data is available, PCFG achieves better performance for all the datasets. This suggests PCFG is *stable*. On the other hand, for all the Markov model based algorithms (JtR-J/B Markov mode, Hashcat Markov mode, OMEN), they do not show such *stability*, i.e., even with more training data, their performance occasionally decreases. For instance, when the training data is increased from 30% to 50%, the percentage of cracked passwords of `Tianya` is reduced from 64.12% to 64.06% for J212, from 65.02% to 65% for J215, and from 12.07% to 9.21% for H4, respectively. Consequently, we believe that *Markov model based password cracking algorithms are easily* **overfitting** *in the training phase while PCFG is stable*.

(ii) No single algorithm is optimal in all the scenarios. According to our results, given 10^9 guesses, OMEN has the best performance when cracking `178.com`, `CSDN`, and `eHarmony`, JtR (J212) has the best performance when cracking `7k7k`, `Hotmail`, `phpBB`, `Renren`, `Rockyou`, and `Tianya`, and PCFG has the best performance when cracking `Gamigo`, `LinkedIn`, `MySpace`, and `Yahoo!`. The size of training data also impacts the performance of each algorithm. For instance, when cracking `Duduniu`, JtR (J212) has the best performance in the 10% and 30% training data scenarios while PCFG has the the best performance in the 50% training data scenario. Consequently, the crackability of a dataset and the performance of an algorithm depends highly on the passwords' structure, composition, and other properties of the target dataset as well as the training data, employed model, and algorithm design. We conclude that *there is no best password cracking algorithm in general scenarios*.

(iii) Different datasets have different crackability. Based on our results, less than 1.5% passwords of *eHarmony* can be cracked by the examined algorithms within $\sim 10^9$ guesses. Although *eHarmony* (a dating site) only requires the password to have 8 or more characters in the registration phase, people tend to choose secure passwords since they may care a lot about protecting their dating information. On the other hand, for some sites (probably low value), e.g., gaming sites, a large portion of their passwords are easily crackable

TABLE 2
Intra-site password cracking. Each value in this table represents the fraction of passwords been cracked in a dataset (e.g., .4826 indicates that 48.26% passwords of a dataset have been cracked). Default number of guesses: $\sim 10^9$ for PCFG, J212, and OMEN; $\sim 1.4 \times 10^9$ for J215 and H4.

	10% **training data**					30% **training data**					50% **training data**				
	PCFG	J212	OMEN	J215	H4	PCFG	J212	OMEN	J215	H4	PCFG	J212	OMEN	J215	H4
17173.com	.4826	.5769	.5711	.5940	.1491	.5776	.5765	.5705	.5934	.1104	.6525	.5771	.5718	.5940	.0829
178.com	.5270	.6028	.6097	.6168	.1839	.5675	.6023	.6096	.6165	.1359	.5828	.6018	.6091	.6161	.1020
7k7k	.4550	.6236	.6024	.6376	.1642	.5849	.6239	.6026	.6379	.1220	.6186	.6243	.6027	.6385	.0914
CSDN	.3312	.3786	.3860	.3941	.1875	.3602	.3774	.3874	.3927	.1386	.3768	.3777	.3866	.3932	.1045
Duduniu	.3731	.4353	.4198	.4571	.0645	.4293	.4366	.4198	.4582	.0478	.4481	.4358	.4209	.4573	.0359
eHarmony	.0068	.0061	.0141	.0073	.0002	.0071	.0063	.0146	.0074	.0002	.0076	.0062	.0142	.0074	.0001
Gamigo	.1130	.1042	.0491	.1127	.0005	.1156	.1042	.0491	.1127	.0003	.1170	.1044	.0492	.1130	.0003
Hotmail	.1728	.4234	.1112	.4359	.0060	.1936	.4497	.2967	.4662	.0054	.2006	.4626	.3240	.4758	.0058
LinkedIn	.1616	.1594	.1333	.1724	.0007	.1636	.1592	.1367	.1721	.0006	.1656	.1589	.1337	.1718	.0004
MySpace	.5150	.4178	.3504	.4401	.0075	.5332	.4258	.4238	.4482	.0060	.5399	.4248	.4407	.4465	.0047
phpBB	.2758	.4271	.3754	.4473	.0032	.2877	.4302	.4176	.4511	.0025	.2921	.4314	.4214	.4523	.0021
Renren	.4090	.5958	.5178	.6116	.1647	.4565	.5962	.5187	.6118	.1219	.4754	.5962	.5177	.6120	.0916
Rockyou	.4623	.5270	.5059	.5445	.0067	.4777	.5265	.5058	.5440	.0050	.4844	.5265	.5055	.5441	.0037
Tianya	.4820	.6408	.5814	.6501	.1654	.5417	.6412	.5815	.6502	.1207	.5633	.6406	.5824	.6500	.0921
Yahoo!	.4050	.3616	.3700	.3797	.0039	.4161	.3594	.3765	.3780	.0032	.4184	.3604	.3797	.3784	.0022

possibly because people just choose easily memorable weak passwords.

(iv) Given the same training data, J215 has better performance than J212. The reason is straightforward since more guesses are made. JtR also has better performance than Hashcat (H4) under our settings. Based on our analysis on the raw results, we believe that the reason is as follows: JtR Markov mode employs an improved technique which enables candidate passwords to be guessed in a decreasing order of possibility; and JtR is better in guessing short passwords (< 9) while Hashcat has better performance in guessing longer passwords (≥ 9) [25]. Meanwhile, based on our statistics in Section 5, more than half of the passwords in most datasets have a length less than 9.

(v) When comparing the results in Fig.1 (training-free, 10^{10} guesses) and Table 2 (intra-site training based, 10^9 or 1.4×10^9 guesses), we can see that the training-based intra-site cracking has better performance in most scenarios even if they make less guesses. For example, when cracking 178.com, JtR-B-Inc can crack 36.02% passwords within 10^{10} guesses while PCFG, J212, and OMEN can crack 52.7%, 60.28%, and 60.97% passwords respectively within 10^9 guesses (10% training data scenario). The reason is as expected: the auxiliary knowledge of passwords is helpful in cracking new passwords.

Training-based Cross-site Cracking. In this subsection, we examine the crackability of the 15 password datasets by conducting a training based cross-site evaluation. The employed cracking algorithms are PCFG, J212, J215, H4, OMEN, and the *semantic based password cracking algorithm* recently proposed in [12], denoted by *Sem* and *Sem+*, where Sem+ makes more guesses than Sem. We consider three scenarios with different training datasets: Tianya, Rockyou, and Tianya+Rockyou. In each scenario, we use the corresponding dataset to train each algorithm, and then use the trained password guesser to crack the 15 datasets (the training dataset is also included in the cracking phase). We summarize the results in Table 3, from which we have the following observations.

(i) As in the training-based intra-site cracking case, no algorithm is optimal in all the scenarios. For instance, to crack 17173.com within $\sim 10^9$ guesses, Tianya-trained Sem cracker has the best performance over other Tianya-trained crackers, Rockyou-trained J212 cracker has the best performance over other Rockyou-trained crackers, while Tianya+Rockyou-trained OMEN cracker has the best performance over other Tianya+Rockyou-trained crackers; similarly, to crack LinkedIn within $\sim 10^9$ guesses, Tianya-trained PCFG cracker and Rockyou-trained Sem cracker are the best compared to their counterparts. Consequently, the actual cracking performance varies depending on not only the cracking algorithm, but also the training data, target data, as well as the number of guesses.

(ii) Different datasets have obvious differences in crackability, e.g., within $\sim 10^9$ guesses, 73.8% MySpace passwords can be cracked by a Rockyou-trained Sem cracker while at most 3.33% eHarmony passwords can be cracked by any of the examined state-of-the-art cracking algorithms. To some extent, these results might reflect how people value their accounts on different websites. We infer that eHarmony, as a dating site, attracts people's attention the most on securing their passwords. A somewhat unexpected observation is the crackability of CSDN, which is a site where most of the users are computer programmers, i.e., the people who are supposed to be more aware of the importance of password security. Based on our results, almost half of its users' passwords can be cracked by a Tianya-trained Sem+ cracker. We believe that the reason is that many CSDN users, even if they are computer programmers, are more likely to consider this site as a forum for technical discussion and resource sharing and thus they do not care about the strength of their passwords.

(iii) *Regional/language difference, i.e., the cultural difference, does affect password crackability.* Generally speaking, as expected, the Tianya-trained crackers are better at cracking Chinese password datasets, e.g., 17173.com, 178.com, 7k7k, CSDN, Duduniu, Renren, while the Rockyou-trained crackers are better at cracking English password datasets, e.g., eHarmony, Hotmail, LinkedIn, MySpace, phpBB, Yahoo!. Furthermore, the Tianya+Rockyou-trained crackers are more robust in cracking both Chinese and English password datasets. This is because Tianya is

TABLE 3
Cross-site password cracking. Each value in this table represents the fraction of passwords been cracked in a dataset (e.g., .4693 indicates 46.93% passwords of a dataset have been cracked). Number of guesses: $\sim 10^9$ for PCFG, Sem, J212, and OMEN; $\sim 1.4 \times 10^9$ for Sem+, J215, and H4.

	Training data: `Tianya`							**Training data: `Rockyou`**							**Training data: `Tianya+Rockyou`**						
	PCFG	Sem	J212	OMEN	Sem+	J215	H4	PCFG	Sem	J212	OMEN	Sem+	J215	H4	PCFG	Sem	J212	OMEN	Sem+	J215	H4
`17173.com`	.4693	.5535	.5524	.5235	.5615	.5677	.0808	.3664	.3632	.4626	.4229	.3696	.4781	.0004	.4732	.5284	.5243	.5334	.5380	.5390	.0811
`178.com`	.4959	.5681	.5522	.5417	.5768	.5674	.0686	.4092	.3931	.4579	.4230	.3988	.4698	.0000	.4974	.5404	.5204	.5409	.5484	.5384	.0707
`7k7k`	.5486	.6179	.6147	.5652	.6248	.6275	.0905	.3807	.3860	.5321	.4816	.3914	.5477	.0006	.5531	.6012	.5943	.5866	.6090	.6076	.0897
`CSDN`	.3790	.4663	.3780	.3817	.4752	.3915	.1047	.2918	.3033	.3015	.3181	.3084	.3163	.0001	.3760	.4314	.3564	.3910	.4414	.3694	.1107
`Duduniu`	.3885	.4801	.3871	.3465	.4920	.4066	.0369	.3214	.3205	.3253	.2863	.3295	.3400	.0007	.3983	.4640	.3716	.3782	.4779	.3890	.0352
`eHarmony`	.0027	.0068	.0027	.0004	.0076	.0029	.0000	.0022	.0333	.0052	.0051	.0381	.0062	.0001	.0025	.0224	.0037	.0026	.0258	.0042	.0000
`Gamigo`	.1158	.0989	.0807	.0405	.1054	.0863	.0003	.1351	.1988	.1300	.1121	.2072	.1388	.0003	.1366	.1844	.1207	.0991	.1926	.1288	.0004
`Hotmail`	.3071	.2839	.2972	.1615	.2936	.3105	.0035	.3279	.5172	.4370	.3778	.5259	.4512	.0030	.3399	.4875	.4143	.3482	.4988	.4293	.0037
`LinkedIn`	.1519	.1332	.0894	.0389	.1426	.0971	.0003	.1776	.2753	.1636	.1343	.2873	.1757	.0004	.1769	.2545	.1479	.1173	.2662	.1589	.0003
`MySpace`	.3648	.4014	.1977	.0619	.4182	.2163	.0015	.5679	.7380	.3913	.3893	.7495	.4146	.0021	.5465	.6884	.3481	.3322	.7026	.3713	.0017
`phpBB`	.3239	.3515	.2855	.1176	.3653	.3007	.0021	.3551	.5470	.4254	.3299	.5602	.4438	.0017	.3609	.5319	.4018	.2949	.5458	.4201	.0022
`Renren`	.5176	.5643	.5683	.4756	.5722	.5823	.0897	.4409	.4903	.5599	.4987	.4968	.5744	.0019	.5328	.6026	.5909	.5602	.6110	.6051	.0869
`Rockyou`	.4190	.4406	.3410	.1651	.4539	.3588	.0186	.5407	.7804	.5266	.5054	.7909	.5441	.0037	.5323	.7317	.4922	.4548	.7434	.5108	.0197
`Tianya`	.7221	.7968	.6405	.5817	.8037	.6499	.0920	.4241	.4281	.5589	.5244	.4322	.5696	.0008	.7193	.7742	.6288	.5971	.7812	.6379	.0937
`Yahoo!`	.3906	.3670	.2061	.0900	.3802	.2203	.0074	.4433	.6023	.3608	.3582	.6138	.3779	.0020	.4416	.5607	.3279	.3161	.5734	.3449	.0081

a Chinese password dataset while `Rockyou` is an English password dataset, and user-chosen passwords are more likely to follow their language/cultural patterns. Consequently, our results are consistent with the findings on Chinese and English password behaviors in [1].

(iv) *The semantics based cracking algorithm Sem/Sem+ is sensitive to regional/language difference.* From Table 3, we can see that `Tianya` and `Rockyou` trained Sem/Sem+ crackers are very sensitive: in most of the scenarios, the `Tianya`-trained Sem/Sem+ cracker performs significantly better in cracking Chinese passwords and the `Rockyou`-trained Sem/Sem+ cracker performs significantly better in cracking English passwords. This is because, unlike other algorithms that do not consider the semantics information, Sem/Sem+ is designed based on the semantics information of passwords. Thus, it is more sensitive to the semantics correlation between the training data and the target data. This further implies that *if the target data and the training data are semantically similar (e.g., they are created by people from the same region, speaking the same language, with similar interests and/or background), the semantics based cracking algorithms are more likely to achieve a better performance*.

(v) When comparing Tables 2 and 3, interestingly, we find that *intra-site cracking does not always have better performance than cross-site cracking even in the* 50% *training data scenario*, e.g., when cracking `Hotmail`, the 50% data-trained PCFG cracker cracks 20.06% of the passwords in intra-site cracking, while `Tianya`-trained, `Rockyou`-trained, and `Tianya+Rockyou`-trained PCFG crackers crack 30.07%, 32.79%, and 33.99% of the passwords in cross-site cracking, respectively. We believe that the reason for this is the *high correlation between training data and target data*.

5 Classification based Analysis

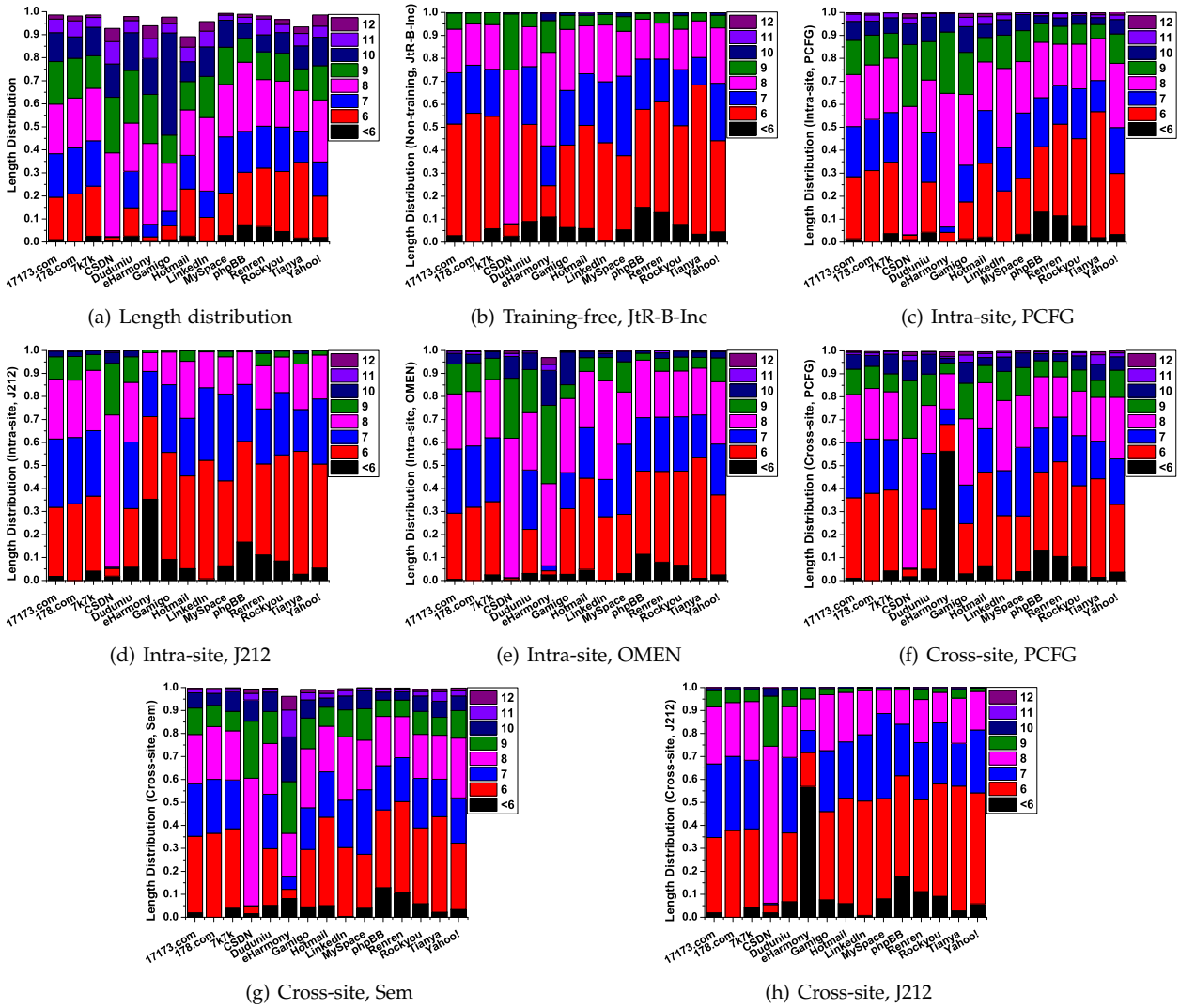
Length based Classification and Evaluation. First, we classify each password dataset and its cracked passwords in terms of *password length* as shown in Fig.2, where (a) shows the length distribution of each original password dataset, (b) shows the length distribution of passwords cracked by JtR-B-Inc (training-free), (c), (d), and (e) show the length distribution of passwords cracked by PCFG, J212, and OMEN (30%-training data-based intra-site cracking) respectively, and (f), (g), and (h) show the length distribution of passwords cracked by PCFG, Sem, and J212 (`Tianya+Rockyou`-trained cross-site cracking) respectively. Note that, in Fig.2, we do not include passwords longer than 12 since they take a small portion of the overall original/cracked passwords. From Fig.2, we have the following observations.

(i) Based on (a), most password datasets consist of passwords of length 6 to 11. Particularly, over 90% of `eHarmony` and `CSDN` users have no passwords with length less than 8, which suggests that they are potentially more secure than other datasets with respect to password length. Actually, according to our cracking results in the previous section, `eHarmony` does have more secure passwords than other datasets while `CSDN` is as secure as other datasets since it is not secure with respect to other criteria.

(ii) From (b), we can see that most of the cracked passwords of JtR-B-Inc have length less than or equal to 9. A similar observation can also be found for J212 in intra-site cracking in (d) and J212 in cross-site cracking in (h). This indicates that JtR is more effective in cracking short passwords than longer passwords. Compared to JtR, PCFG (both intra-site and cross-site cracking scenarios), OMEN, and Sem can crack a larger number of longer passwords with length no less than 9, which can be observed from (c), (e), (f), and (g).

(iii) If we compare PCFG/Sem with JtR/OMEN, we find the structure based password cracking algorithms (including PCFG and semantics based algorithms) are more powerful in cracking longer passwords (length ≥ 9) than Markov model based algorithms. The reason is that: structure based algorithms generate password guesses according to the structure of training passwords, i.e., if the training data consists of more longer passwords, more guesses with longer length will be generated; while Markov model based algorithms generate guesses according to a trained Markovian state graph, and thus longer password guesses have lower probability to be generated than shorter passwords.

(iv) In summary, we conclude that *generally speaking, if no other auxiliary information is present, e.g., expired password history, longer passwords are more secure than shorter passwords.* Therefore, the requirement on the minimum length of user-chosen passwords in many password policies has a positive influence in securing users' passwords.



(a) Length distribution (b) Training-free, JtR-B-Inc (c) Intra-site, PCFG

(d) Intra-site, J212 (e) Intra-site, OMEN (f) Cross-site, PCFG

(g) Cross-site, Sem (h) Cross-site, J212

Fig. 2. Length distribution of original and cracked passwords.

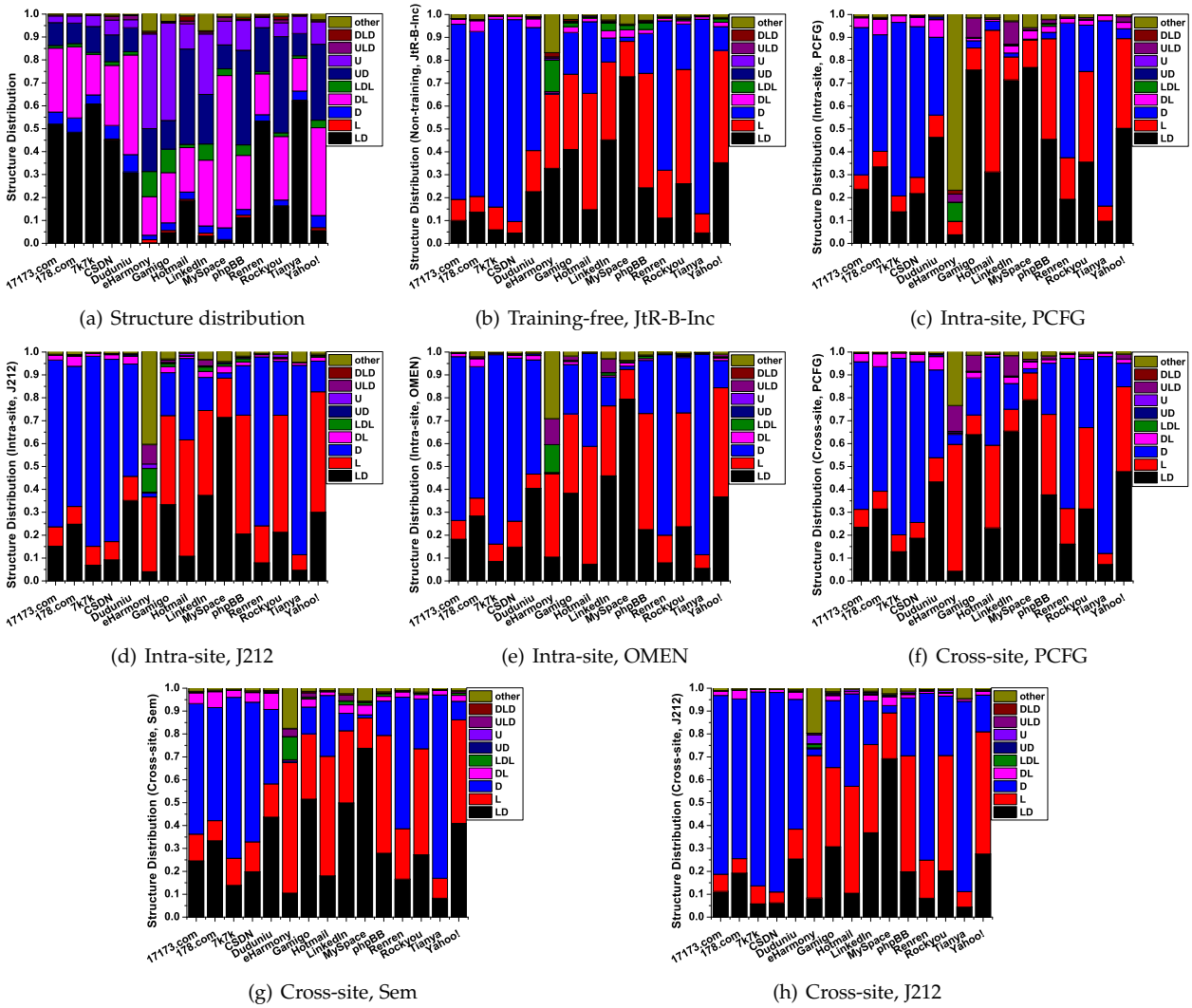
Structure based Classification and Evaluation. We now classify the 15 password datasets and the cracked passwords of each dataset based on their structure as shown in Fig.3, where (a) shows the structure distribution of each original dataset, (b) shows the structure distribution of passwords cracked by JtR-B-Inc (training-free), (c), (d), and (e) show the structure distribution of passwords cracked by PCFG, J212, and OMEN respectively in 30%-training data-based intra-site cracking, and (f), (g), and (h) show the structure distribution of passwords cracked by PCFG, Sem, and J212 respectively in `Tianya+Rockyou`-training-based cross-site cracking. When conducting the structure based classification, we consider 9 popular password structures: LD, L, D, DL, LDL, UD, U, ULD, and DLD [1][2]. We classify other password structures as "other". From Fig.3, we have the following observations.

(i) From (a), all the Chinese password datasets have a significant portion of passwords with the LD structure. Furthermore, the Chinese password datasets have more digit-only passwords than English and German password datasets, which is consistent with the observation in [1]. Furthermore, we can find that most passwords of `eHarmony` and `Gamigo` have structures of LDL, UD, U, and other unpopular structures. These structures are relatively more secure compared to structures LD, L, D, and DL. Consequently, `eHarmony` and `Gamigo` are more difficult to crack, which is consistent with our previous results.

(ii) With respect to structure, `Tianya` and `Rockyou` have similar distributions of Chinese and English password datasets, respectively. This explains why the `Tianya`-trained crackers are more powerful in cracking Chinese passwords while `Rockyou`-trained crackers are more powerful in cracking English passwords.

(iii) From (b)-(h), we can see that most of the passwords cracked by the considered algorithms have structures of LD, L, and D, which are relatively simple and very popular password structures. Particularly, we can see that most of the cracked Chinese passwords have a structure D while most of the cracked English passwords have two structures L and LD. Therefore, if the language information of the target dataset is available, proper training data and algorithm can be chosen to achieve better cracking performance.

(iv) By comparing PCFG and Sem in the cross-site cracking scenario, we find that Sem is more effective than



(a) Structure distribution (b) Training-free, JtR-B-Inc (c) Intra-site, PCFG

(d) Intra-site, J212 (e) Intra-site, OMEN (f) Cross-site, PCFG

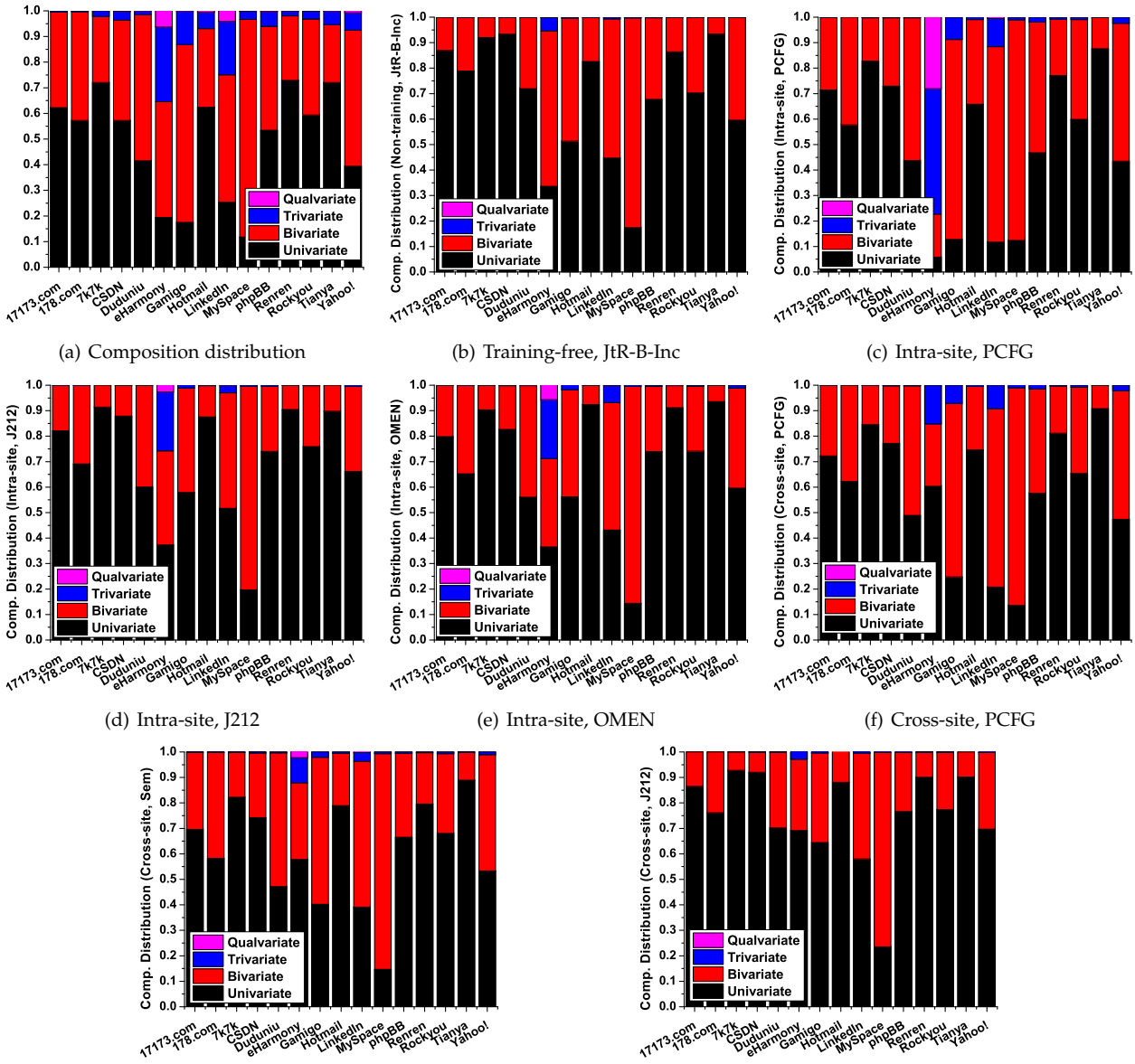
(g) Cross-site, Sem (h) Cross-site, J212

Fig. 3. Structure distribution of original and cracked passwords.

PCFG in cracking letter based passwords, e.g., passwords with structure L. This is because PCFG only considers the password structure information while overlooking the letter part, while Sem improved PCFG by involving both password structure and semantics information into consideration.

Composition based Classification and Evaluation. Now, we classify each password dataset and the cracked passwords based on password composition. First, we assume passwords are composed of four sets of symbols: L, U, D, and S. Then, in terms of the four sets of symbols, we classify passwords into four categories: *univariate*, *bivariate*, *trivariate*, and *qualvariate* passwords, which consist of passwords composed by one, two, three and four set(s) of symbols, respectively. We illustrate the results in Fig.4, where (a) shows the composition distribution of each dataset, (b) shows the composition distribution of passwords cracked by JtR-B-Inc in training-free cracking, (c), (d), and (e) show the composition distribution of passwords cracked by PCFG, J212, and OMEN respectively in 30%-training data-based intra-site cracking, and (f), (g), and (h) show the composition distribution of passwords cracked by PCFG, Sem, and J212 respectively in `Tianya+Rockyou`-trained cross-site cracking.

From Fig.4, we have the following observations. (i) Most of the considered datasets consist of univariate and bivariate passwords, which implies that to crack a large portion of a password dataset, the search space for a cracking algorithm is significantly reduced. This enables researchers to design more effective password cracking algorithms. In other words, this fact provides the foundation of the success of modern password cracking algorithms, e.g., JtR, PCFG, OMEN, Sem. Specifically, we notice that `eHarmony` has the largest portion of Trivariate and Qualvariate passwords, which are relatively secure passwords. This implies `eHarmony` is the one most difficult to crack, which is consistent with our evaluation results. (ii) Based on (b)-(h), most of the cracked passwords are univariate or bivariate passwords. The reasons are as follows: they are relatively simple with respect to composition; they come from a smaller password space; more training data with similar composition are available, which enables the cracking algorithms to train a more accurate cracker. (iii) Our results also agree with the requirement that *the chosen password should contain characters from three or more symbol sets* in some password policies. Although this requirement could reduce



(a) Composition distribution (b) Training-free, JtR-B-Inc (c) Intra-site, PCFG

(d) Intra-site, J212 (e) Intra-site, OMEN (f) Cross-site, PCFG

(g) Cross-site, Sem (h) Cross-site, J212

Fig. 4. Composition distribution of original and cracked passwords.

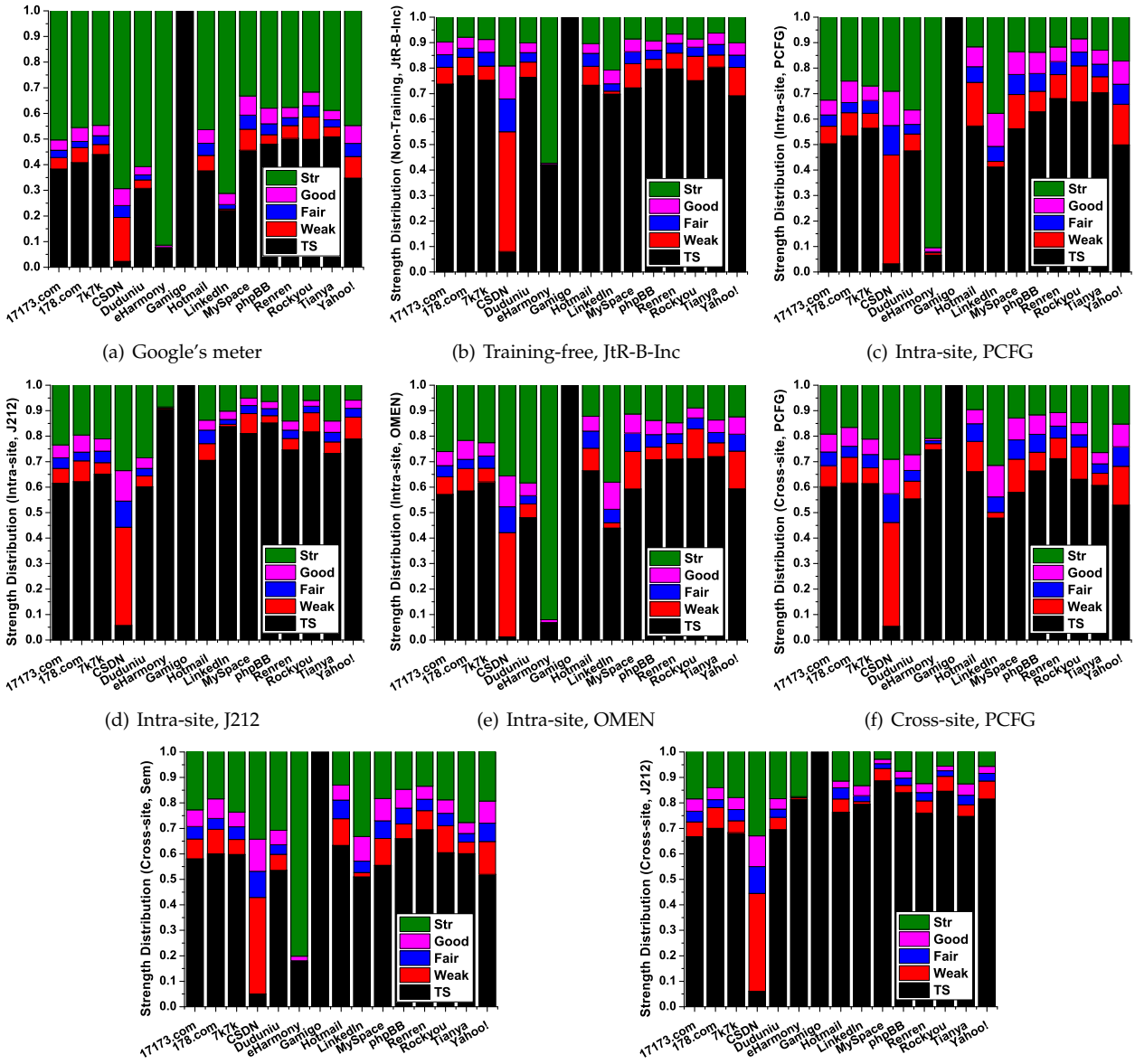
the memorability of qualified passwords, it can significantly improve the password security.

6 Commercial Password Meters

In this section, we examine the effectiveness and soundness of commercial password meters, as well as their impacts on password security. The motivation of this part of the evaluation is to try to understand two questions: *how commercial password meters evaluate and classify the strength of passwords? and how the commercial password meters affect password security?* To address the first question, we improved the *password strength testing tool* implemented by Carnavalet and Mannan in [22]. The improved integrated testing tool is named *Automatic Password strength Testing tool* (AutoPassTest). AutoPassTest can automatically send the testing passwords to the sever of a commercial password meter for evaluation, and return the evaluation results back to the client. Furthermore, when testing a large number of passwords, AutoPassTest can restart automatically and dynamically adjust the testing frequency to avoid causing too much traffic and load on the sever. To address the second question, we will evaluate the strength of the cracked passwords in terms of commercial password meters.

In our evaluation, we conduct $\sim$ 600 million password-strength-tests using the 15 password datasets on four popular sites: Google (English), Twitter (English), QQ (Chinese), and 12306.cn (Chinese). Due to the space limitations, we only show the results of Google's meter and QQ's meter in Fig.5 (a) and Fig.6 (a), respectively.

From Fig.5 (a) and Fig.6 (a), we have the following observations. (i) Interestingly, Google's meter classifies most passwords of each dataset as either *Strong* or *Too Short* even though it has five levels, which implies that for a password of a dataset, if it is accepted by Google's meter (not *Too*



(a) Google's meter (b) Training-free, JtR-B-Inc (c) Intra-site, PCFG

(d) Intra-site, J212 (e) Intra-site, OMEN (f) Cross-site, PCFG

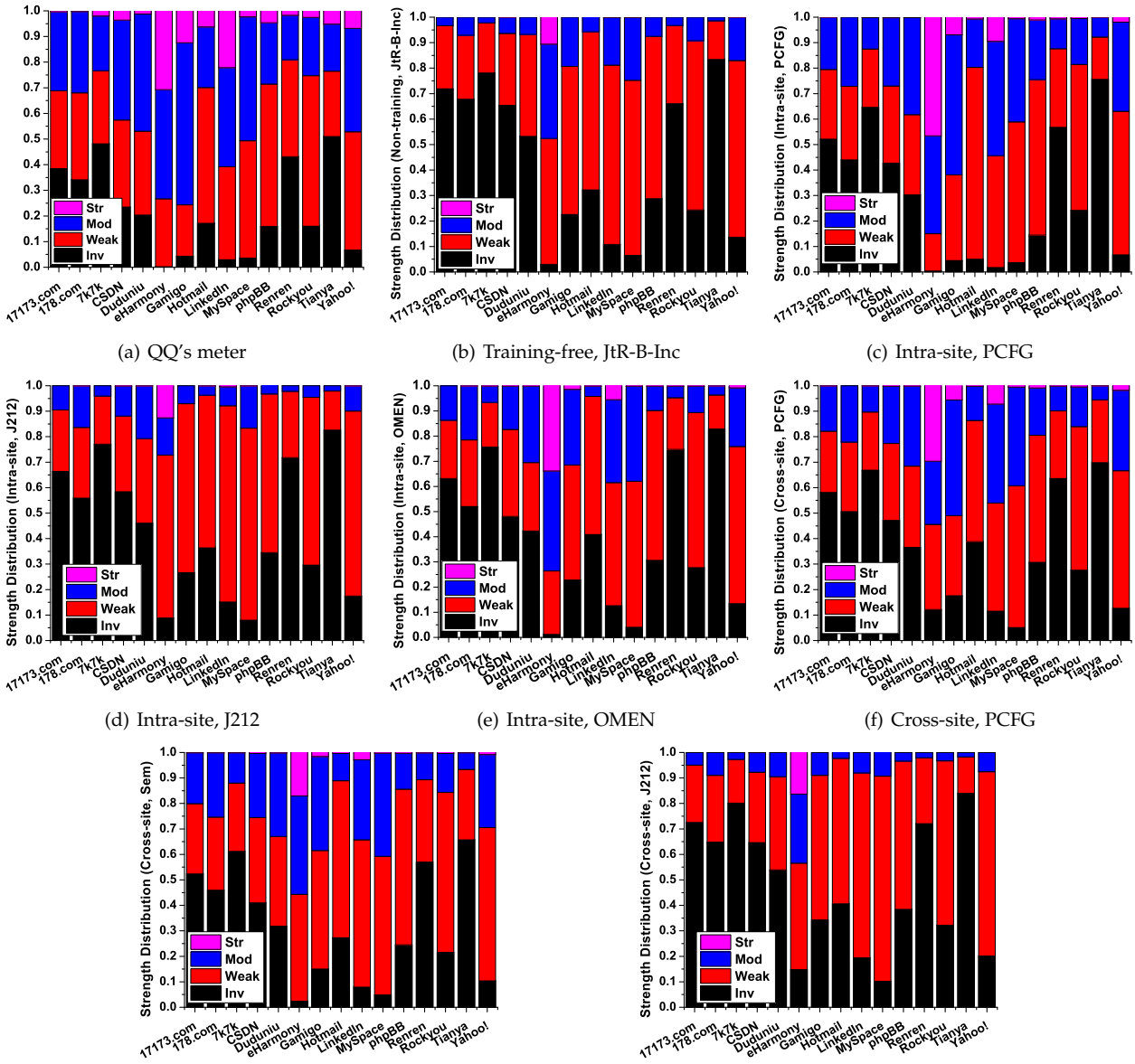
(g) Cross-site, Sem (h) Cross-site, J212

Fig. 5. Google's meter based strength distribution of original and cracked passwords. **TS** = *Too Short* and **Str** = *Strong*.

Short), then it is very likely a strong password with respect to Google's meter. (ii) The classification results of QQ's meter are very different from that of Google, since most passwords are classified as either *Weak* or *Moderate*. This suggests that QQ's meter is more cautious than Google's meter. (iii) Both Google's meter and QQ's meter agree that `eHarmony` has more secure passwords than other datasets. This implies that `eHarmony` is potentially more difficult to crack, which is consistent with our evaluation results.

Now, we employ Google's meter and QQ's meter to evaluate the strength of the cracked passwords in Section 4 in Fig.5 (b)-(h) and Fig.6 (b)-(h), respectively, where in both figures, (b) shows the strength classification of the cracked passwords of JtR-B-Inc in the training-free cracking scenario, (c), (d), and (e) show the strength classification of the cracked passwords of PCFG, J212, and OMEN respectively in the 30%-training data-based intra-site cracking, and (f), (g), and (h) show the strength classification of the cracked passwords of PCFG, Sem, and J212 respectively in the `Tianya+Rockyou`-training based cross-site cracking.

First, from Fig.5 (b)-(h), we have the following observations. (i) For most cracked passwords, they are classified as either *Too Short* or *Strong* by Google's meter. It is not a surprise that the "*Too Short*" passwords were cracked. However, it is unexpected that many "*Strong*" passwords are also relatively easily crackable. We believe the reasons are: *Google's meter may not be very accurate in classifying user-chosen passwords, i.e., even if a password gets a strength rating as "Strong", it is still not as secure as the rating implies;* and *the modern password crackers, especially the trained crackers, are powerful. Since the trained data also has Strong passwords, the cracker can generate proper guesses based on the trained knowledge.* (ii) For `CSDN` passwords, we can see that most of its passwords are acceptable by Google's meter. Unfortunately, from the



(a) QQ's meter (b) Training-free, JtR-B-Inc (c) Intra-site, PCFG

(d) Intra-site, J212 (e) Intra-site, OMEN (f) Cross-site, PCFG

(g) Cross-site, Sem (h) Cross-site, J212

Fig. 6. QQ's meter based strength distribution of original and cracked passwords. **Inv** = *Invalid*, **Mod** = *Moderate*, and **Str** = *Strong*.

cracking results in Section 4 and the classification in Fig.5, a significant portion of CSDN passwords can be cracked even if they are classified as *Strong* or *Good*. (*iii*) By comparing the results in Fig.5 (c)-(h), we find that PCFG, Sem, and OMEN have better performance in cracking *Good* and *Strong* passwords than JtR Markov mode. We believe that the reason for this is that PCFG, Sem, and OMEN are more effective in training their crackers. Especially PCFG and Sem, since they all consider the structure probability, they are more effective in generating *Good* or *Strong* password guesses if similar passwords appear in the training data. Therefore, if the target is to crack more strong passwords, PCFG, Sem, and OMEN may achieve better performance (an implicit assumption is that the training data should also include strong password instances).

Now, we analyze the results in Fig.6 (b)-(h), from which we have the following observations. (*i*) As expected, most cracked passwords are classified as *Invalid*, *Weak*, or *Moderate* by QQ's meter, and only a small percentage of cracked passwords are labeled as *Strong*. Therefore, with respect to the considered datasets and cracking algorithms, the classification results of QQ's meter are more appropriate (reasonable) than that of Google's meter. This also implies that a proper password meter is helpful in guiding users to choose secure passwords. (*ii*) Again, from Fig.6 (c)-(h), PCFG, Sem, and OMEN are better than JtR in cracking *Moderate* and *Strong* passwords. The reason is the same as shown in the previous analysis.

In summary, we conclude that *(i) some password meters (e.g., Google's meter) cannot classify passwords at their appropriate strength level, which implies a simple modification (by adding one digit) on unacceptable or weak passwords could turn these passwords into good or strong ones; (ii) some password meters' classification results are not very useful, i.e., even if the passwords*

TABLE 4
Average number of guesses for each username/email. % = *fraction of passwords been cracked* and # = *average guess numbers*.

Username			Email					
Dataset	%	#	Dataset	%	#	Dataset	%	#
17173	.6146	364	17173	.5282	626	MySpace	.4976	1424
178	.5325	531	7k7k	.6236	527	Renren	.6175	797
CSDN	.4611	789	CSDN	.4146	1130	Tianya	.6230	630
Tianya	.6429	392	Duduniu	.5324	706	Yahoo!	.4297	2569

are classified as strong or good, they may still be vulnerable to modern password cracking algorithms. Consequently, these inept password meters may confuse or even mislead users when choosing passwords; and (iii) there are some relatively accurate password meters, which are useful in helping users choose secure passwords against modern password cracking algorithms.

7 Impacts of Username/Email Leakage

In some password leakage incidents, the associated usernames and emails are also leaked along with the passwords. For instance, among our considered 15 leaked password datasets, 17173.com, 178.com, CSDN, and Tianya have username information available, and 17173.com, 7k7k, CSDN, Duduniu, MySpace, Renren, Tianya, and Yahoo! have email information available. Intuitively, the username and email information have security impacts on password security since people may follow the same/similar style in choosing their usernames/email aliases and passwords. In this section, we examine the security impacts of usernames and email addresses on passwords[2].

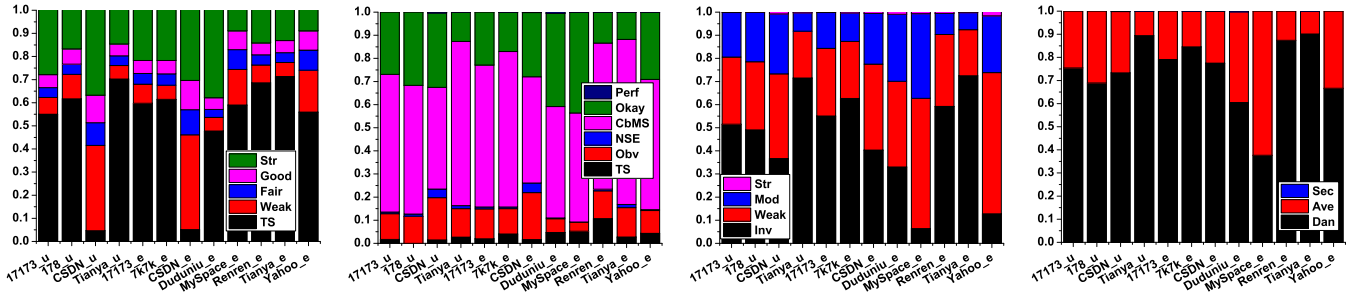
The employed password cracking algorithm is *JtR single crack mode* (JtR-J and JtR-B has the same performance in the single crack mode according to our results), denoted by *JtR-Sin*. Under JtR-Sin, each target password is associated with an username/email. Then, multiple *mangling rules* will be applied to the username/email to generate password guesses [13][14], e.g., some digits may be added to a string username/email (from username/email alias "*softquery*" to password guesses "*softquery1*", "*softquery11*"), switch the first lower case letter to its upper case form, (from username/email alias "*aswind*" to password guess "*Aswind*").

We summarize the username and email based password crackability and the average number of guesses made based on each username/email alias in Table 4, from which we have the following observations. (i) Both username leakage and email leakage do have a surprising impact on password security. Based on username information, 61.46%, 53.25%, 46.11%, and 64.29% passwords of 17173.com, 178.com, CSDN, and Tianya can be cracked respectively within only 364 to 789 guesses on average. Similarly, the email information is also very powerful in cracking people's passwords. For instance, 62.36%, 49.76%, and 61.75% passwords of 7k7k, MySpace, and Renren can be cracked respectively within only 527-1424 guesses on average. This is a serious alert to password users and system administrators. When evaluating password-based authentication systems' security, besides evaluating the strength of passwords themselves, the correlation between passwords and usernames, emails, and other personal information should be considered. We believe the obtained results can give insight to password meter designers, i.e., the users' profiles should be involved in evaluating user-chosen passwords[3]. (ii) Based on the results, username information is more powerful than email information in cracking passwords. For instance, based on username information, 61.46% passwords of 17173.com can be cracked within 364 guesses, 46.11% passwords of CSDN can be cracked within 789 guesses, and 64.29% passwords of Tianya can be cracked within 392 guesses; while based on email information, 52.82% passwords of 17173.com can be cracked within 626 guesses, 41.46% passwords of CSDN can be cracked within 1130 guesses, and 62.3% passwords of Tianya can be cracked within 630 guesses, i.e., username based cracking achieves better performance with less guesses. We believe that the reason for this is as follows: when people register an account, they usually use existing email accounts while choosing new usernames and passwords. Consequently, it is more likely that usernames follow more similar structural, syntactic, and semantic patterns with passwords.

Now, we evaluate the strength of the cracked passwords in Table 4 with respect to popular commercial password meters, Google, Twitter, QQ, and 12306.cn. The results are shown in Fig.7, where "*_u" and "*_e" denote the username based and email based cracking scenarios, respectively. From Fig.7, we have the following observations. (i) For most of the username/email based cracking results, Twitter's meter classifies them as *Obvious*, *Could be More Secure*, and *Okay*, QQ's meter classifies them as *Invalid*, *Weak*, and *Moderate*, and 12306.cn's meter classifies them as *Dangerous* and *Average*. The strength classification results are relatively normal compared with Google's meter, since from (a), we can see that a significant portion of cracked passwords are also labeled as *Strong* by Google's meter. This implies: *again, Google's meter is not as useful as expected; and Twitter's meter, QQ's meter, and 12306.cn's meter, especially 12306.cn's meter, are helpful in guiding users to choose secure passwords against various attacks.* (ii) From Figs. 5, 6, and 7, although the username/email based cracking is more powerful in cracking relatively low-strength passwords, which is similar to the traditional model based algorithms (e.g., PCFG, OMEN, Sem), it has its own advantage: since for each password, it only has to make a small number of tries ($\sim 364-2569$ guesses) based on the username/email, which implies it is faster. Therefore, the username/email based password cracking technique should also be involved in a future promising hybrid cracking strategy.

2. Note that, in [16], Dürmuth et al. also implemented a user profiles based cracking scheme OMEN+. OMEN+ incorporates users' profiles (first name, education, occupancy, birthday) into password cracking. The results based on 3410 passwords together with corresponding user profiles demonstrated that up to 5% cracking performance improvement can be achieved. Since the user profiles of the considered datasets (which are much larger than the dataset in [16]) are not available for us, we do not evaluate OMEN+ in this paper.

3. Although some existing password meters have considered usernames when evaluating the entered password's strength, they usually simply alert users not to choose passwords same as usernames. Furthermore, we suggest other personal information be included in the design of password meters.



(a) Google's meter (b) Twitter's meter (c) QQ's meter (d) 12306.cn's meter

Fig. 7. Strength evaluation of username/email based cracking results. **TS** = *Too Short*, **Str** = *Strong*, **Obv** = *Obvious*, **NSE** = *Not Secure Enough*, **CbMS** = *Could be More Secure*, **Perf** = *Perfect*, **Inv** = *Invalid*, **Mod** = *Moderate*, **Dan** = *Dangerous*, **Ave** = *Average*, and **Sec** = *Secure*.

8 Discussion, Limitations, and Future Work

Ethical Considerations. In our evaluation, we employ 15 leaked password datasets ($\sim$ 145 million passwords). All of the datasets are publicly available now, and have been used extensively in password analysis and research [1]-[23]. However, we also realize these datasets were initially obtained and published illegally. As in [1], [2], [3], [5], [12], [16], [19], we use these data only for research.

Hybrid Password Cracking. Based on our evaluation results, some algorithms, e.g., JtR-B-Inc, JtR-J-Inc, Hashcat-Mask, are training-free while some other powerful algorithms, e.g., PCFG, Sem, JtR Markov mode, OMEN, require priori data for training the cracker. Meanwhile, some algorithms, e.g., PCFG, Sem, OMEN, have early-development advantages (their initial password guesses are effective) while some other algorithms, e.g., JtR Markov mode, have late-development advantages. Therefore, the evaluation results imply that a *hybrid password cracking strategy* could be a promising idea to improve password cracking performance. This is still an open research area.

Password Correlation. Based on our results, passwords' regional/language (cultural, in other words) differences can be observed. This finding sheds light on selecting the training data when cracking a target dataset. Besides the need to account for cultural differences when training, this finding also has other implications. First, since passwords do have correlation, the leakage of one password dataset will have impacts to the security of other password datasets. Second, the finding can shed light on password meter research. Most current meters usually evaluate an input password based on the password itself. Since passwords do have correlations, the strength evaluation could be improved if such correlation is considered.

Users' Profile and Password Security. Based on our results, usernames and emails have serious impacts on password security. Furthermore, it is possible that other user profiles also have security impacts on passwords [16]. To make things worse, it is not difficult to crawl users' profile data online in large scale [16]. Therefore, in addition to protecting users' passwords, it is also important to protect other information associated with users. Furthermore, this has implications on designing proper password meters. When evaluating the strength of an input password, user-profile based evaluation could be more sound.

Password Meters. As shown by our evaluations results, different password meters may have very different impacts on the strength of user-chosen passwords. The inconsistency of different password meters has already been observed [22]. Besides that, for some password meters, the passwords labeled as strong by the meter may not be as secure as expected. From this point of view, flawed password meters make things worse with respect to password security. On the other hand, as expected, there are some password meters that can properly evaluate passwords' strength against modern password cracking algorithms. Therefore, they can help users choose more secure passwords. As indicated in our previous discussion, it is expected that current password meters will be improved by evaluating password correlation and by considering users' profile information.

Limitations. In this paper, we focus on evaluating the automated offline password-crack attack. We do not consider phishing attacks, online attacks, or shoulder surfing attacks to passwords.

Future Work. First, we propose to evaluate more password datasets versus cracking algorithms. Second, we will address the challenges of designing a *hybrid password cracking strategy* and propose to develop robust, sound, and efficient hybrid password cracking algorithms. Third, we will study how to properly involve password correlation knowledge and users' profiles into the design of more accurate password meters. Finally, we plan to develop a *uniform* and *open-source* password security evaluation system, which enables users, administrators, and researchers to conduct comprehensive and comparative password security evaluation and analysis.

9 Conclusion

In this paper, we conducted a large-scale empirical study on the crackability, correlation, and security of 15 real world password datasets, which consist of $\sim$ 145 million passwords and cover various popular Internet services and applications. Subsequently, we examined the effectiveness and soundness of commercial password meters and the security impacts of usernames and emails on passwords. Finally, we discussed the implications and limitations of our results and findings in this paper. Our evaluation is expected to help both password users and system administrators understand the vulnerability of current passwords and shed light on future password research.

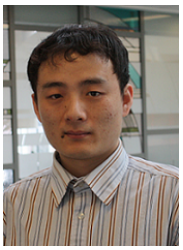
ACKNOWLEDGMENT

The authors are very grateful to Rafavel Veras, Markus Dürmuth, Saranga Komanduri, and Tielei Wang for the help. Weili Han's work was partly supported by the National Science of Foundation of China (NSFC) under grant No. 61572136.

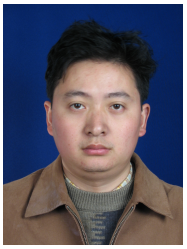
REFERENCES

[1] Z. Li, W. Han, and W. Xu, *A Large-Scale Empirical Analysis on Chinese Web Passwords*, Usenix Security 2014.
[2] J. Ma, W. Yang, M. Luo, and N. Li, *A Study of Probilistic Password Models*, S&P 2014.
[3] M. Dell' Amico, P. Michiardi, and Y. Roudier, *Password Strength: An Empirical Analysis*, Infocom 2010.
[4] D. Florêncio and C. Herley, *A Large-Scale Study of Web Password Habits*, WWW 2007.
[5] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, *The Tangled Web of Password Reuse*, NDSS 2014.
[6] Y. Zhang, F. Monrose, and M. K. Reiter, *The Security of Modern Password Expiration: An Algorithmic Framework and Empirical Analysis*, CCS 2010.
[7] D. Florêncio and C. Herley, *Where Do Security Policies Come From?*, SOUPS 2010.
[8] J. Bonneau, *The Science of Guessing: Analyzing an Anonymized Corpus of 70 Million Passwords*, S&P 2012.
[9] M. L. Mazurek, S. Komanduri, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, P. G. Kelley, R. Shay, and B. Ur, *Measuring Password Guessability for an Entire University*, CCS 2013.
[10] A. Narayanan and V. Shmatikov, *Fast Dictionary Attacks on Passwords using Time-Space Tradeoff*, CCS 2005.
[11] M. Weir, S. Aggarwal, B. Medeiros, and B. Glodek, *Password Cracking Using Probabilistic Context-Free Grammars*, S&P 2009.
[12] R. Veras, C. Collins, and J. Thorpe, *On the Semantic Patterns of Passwords and their Security Impact*, NDSS 2014.
[13] John the Ripper 1.7.9-jumbo-7, *http://www.openwall.com/john/*.
[14] John the Ripper-bleeding-jumbo, *https://github.com/magnumripper/JohnTheRipper*.
[15] Hashcat v0.47, *http://hashcat.net/hashcat/*.
[16] M. Dürmuth, A. Chaabane, D. Perito, and C. Castelluccia, *When Privacy meets Security: Leveraging Personal Information for Password Cracking*, CoRR abs/1304.6584, 2013.
[17] M. Weir, S. Aggarwal, M. Collins, and H. Stern, *Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords*, CCS 2010.
[18] C. Castelluccia, M. Dürmuth, and D. Perito, *Adaptive Passwords-Strength Meters from Markov Models*, NDSS 2012.
[19] P. G. Kelley, S. Komanduri, M. L. Mazurek, R. Shay, T. Vidas, L. Bauer, N. Christin, L. F. Cranor, and J. López, *Guess Again (and Again and Again): Measuring Password Strength by Simulating Password-Cracking Algorithms*, S&P 2012.
[20] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer, N. Christin, and L. F. Cranor, *How Does Your Password Measure Up? The Effect of Strength Meters on Password Creation*, USENIX 2012.
[21] S. Ji, S. Yang, T. Wang, C. Liu, W.-H. Lee, and R. Beyah, *PARS: A Uniform and Open-source Password Analysis and Research System*, ACSAC 2015.
[22] X. C. Carnavalet and M. Mannan, *From Very Weak to Very Strong: Analyzing Password-Strength Meters*, NDSS 2014.
[23] J. Bonneau, C. Herley, P. C. Oorschot, and F. Stajano, *The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes*, S&P 2012.
[24] Keyboard_dic, *https://sites.google.com/site/reusablesec/Home/custom-wordlists*.
[25] *http://www.adeptus-mechanicus.com/codex/jtrhcmkv/jtrhcmkv.php*.
[26] *http://www.zdnet.com/blog/security/chinese-hacker-arrested-for-leaking-6-million-logins/11064*.
[27] *http://www.darkreading.com/attacks-and-breaches/yahoo-hack-leaks-453000-voice-passwords/d/d-id/1105289?*.

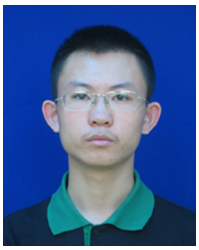
Shukun Yang is currently pursuing his M.S. degree in the School of Electrical and Computer Engineering at Georgia Institute of Technology. He received his B.S. degree from the School of Electrical and Computer Engineering at Georgia Institute of Technology. His research interests include passwords and Statistical Machine Learning. He is a student member of ACM and IEEE.



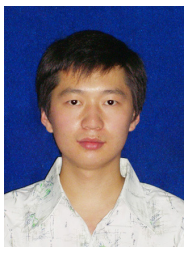
Xin Hu is currently a Research Scientist in the Security Services (GSAL) Team of IBM Security Research Department at IBM Thomas J. Watson Research Center . Before joining IBM, He obtained his Ph.D. in Computer Science and Engineering at the University of Michigan, Ann Arbor in 2011.



Weili Han is an associate professor at Fudan University. His research interests are mainly in the fields of Access Control, Digital Identity, IoT security. He is now the members of the ACM, SIGSAC, IEEE, and CCF. He received his Ph.D. at Zhejiang University in 2003. Then, he joined the faculty of Software School at Fudan University. From 2008 to 2009, he visited Purdue University as a visiting professor funded by China Scholarship Council and Purdue University. He serves in several leading conferences and journals as PC members, reviewers, and an associate editor.



Zhigong Li is pursuing his M.S. in Computer Science at the Fudan University. His research interest is Password Security.



Shouling Ji is a Ph.D. student in the School of Electrical and Computer Engineering at Georgia Institute of Technology. His current research interests include Big Data Security and Privacy, Differential Privacy, Password Security, and Machine Learning Security and Privacy. He also has interests on Graph Theory, and Wireless Networks. He is now an student member of ACM, IEEE, and IEEE COMSOC.

Raheem Beyah is an Associate Professor in the School of Electrical and Computer Engineering at Georgia Tech. His research interests include network security, wireless networks, network traffic characterization and performance, and security visualization. He received the National Science Foundation CAREER award in 2009. He is a member of NSBE, ASEE, and a senior member of ACM and IEEE.