Exploring the Network of Real-World Passwords: Visualization and Estimation

Xiujia Guo[1(✉)], Zhao Wang[1,2], and Zhong Chen[1,2]

[1] School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China
{guoxj,wangzhao,zhongchen}@pku.edu.cn
[2] Key Laboratory of High Confidence Software Technologies, Ministry of Education, Peking University, Beijing 100871, China

Abstract. The distribution of passwords has been the focus of many researchers when we come to security and privacy issues. In this paper, the spatial structure of empirical password sets is revealed through the visualization of disclosed password sets from the website of hotmail, 12306, phpbb and yahoo. Even though the choices of passwords, in most of the cases, are made independently and privately, on closer scrutiny, we surprisingly found that the networks of passwords sets of large scale individuals have similar topological structure and identical properties, regardless of demographic factors and site usage characteristics. The visualized graph of passwords is considered to be a scale-free network for whose degree distribution the power law is a good candidate fit. Furthermore, on the basis of the network graph of the password set we proposed, the optimal dictionary problem in dictionary-based password cracking is demonstrated to be equivalent in computing complexity to the dominating set problem, which is one of the well-known NP-complete problems in graph theory. Hence the optimal dictionary problem is also NP-complete.

Keywords: Computer security · Visualization · Password sets
Power-law distribution · Scale-free network · NP-complete

1 Introduction

Textual password has been a ubiquitous way to access resources and web services since 1960s and the attempts of password cracking have never stopped ever since. Especially in recent years, the leakage of massive password sets repeatedly reminds us of the urgency of password sets security enhancement. While at the same time, what we can do or what we have done to protect the privacy of users and to assure the security of the system seems plausible but far-fetched. Why? As some researchers pointed out, users remain to be the weakest part of the whole password security system and the answer lies in password itself.

While different password cracking techniques have been adopted in prior works, dictionary based password cracking remains to be the most common way

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2018
X. Lin et al. (Eds.): SecureComm 2017, LNICST 238, pp. 147–166, 2018.
https://doi.org/10.1007/978-3-319-78813-5_8

in numerous attacks nowadays. Conventional dictionary based password cracking techniques, such as the statistical guessing attack, usually start with a preprocessed dictionary and might involve some modification during the guessing process. Related research has been made by MSRA [1]. While due to the variations of original data set and dictionary size, the performance differs from one to the other. Bonneau made the first comparison in [2]. Nevertheless, dictionary based cracking techniques were proved effective and feasible in practice.

Dictionary based password cracking technique was first proposed by Morris and Thompson in their seminal analysis of 3,000 passwords in 1979 [3], and the two approaches, password cracking and semantic evaluation, were widely used ever since, even after Markov and PCFG based password crackers were introduced. Even though dictionary based cracking techniques distinguish themselves with feasible performance in practice and play a role of benchmark in a variety of password cracking implements, the reason why they work well remains unknown.

Meanwhile, even though a great deal of works have been done on password creation policy and password strength meters, the gap between our understanding of the security of one single password and the security of a whole password set was rarely discussed. To prevent a password from being compromised, prior works have focused on two metrics: improving the strength of one single password and blocking out passwords whose usage frequency exceed a particular threshold, which is intuitively reasonable but far from perfect.

For the former approach of assuring security, the first question is the definition of strong passwords, i.e. how to measure the security of a password and how to decide whether a password is strong or weak. Bonneau made a survey of related literature and proposed the concept of guessing entropy, α-guess-work [2]. Common practice is the requirement of the length and the variety of characters in a password, such as having at least 8 characters, one lower case character, one capital letter and one number, etc.

For the latter approach of maintaining a blacklist of popular passwords, it seems to be a game of cat and mouse. For every password that is blocked, the user could always make a way out by performing a minor modification on it, for instance, by adding some characters at the rear, changing one or two digits, switching the first character into upper case, or simply using some other weak password that is not included in the list. The minor modifications not only make the blacklist useless, but also leave a potential threat to the entire system. For the same blacklist, if everyone makes his or her own minor modification based on a group of popular passwords, the results could be different but similar to each other. For example, if we all submit "password" as our password and it was blocked, the possible choices after minor modification might be "password1", "password12", "password123", "p@ssword", "Password", etc. As we will discuss in this study, the leakage of one single vulnerable password could lead the compromise of password one after another, thus creating a chain reaction and endanger numerous accounts.

Our first contribution is the visualization of several empirical password sets including the leakage of 12306 (the official website of China Railway Customer

Service Center), hotmail, phpbb and yahoo[1]. We build networks based on the interconnection of the passwords. To our knowledge, this is the first visualization of large scale password sets in the form of networks. Through the graph of the data, we reveal what the topological structure of a whole password set is like in the complete password space.

The second contribution is the exploration of the spatial structure of the data sets we have. The discussion will shed some light on the distribution of passwords, which has been the concern for many years. Malone and Maher [4] investigated frequency distributions of passwords, they pointed that rather than a theoretically desirable uniform distribution, Zipf model usually provides better predictions than a simple uniform model. Malone et al. claimed that the Zipf's Law is a good candidate for modeling the frequency of users-chosen passwords. While the frequency of passwords only indicates the distribution of identical passwords, in this paper, our results support the claim that the visualized graph of passwords is a scale-free network, because the power law distribution is a good estimation of the degree distribution of a password set's visualized graph. Unlike the frequency of passwords, the degree distribution indicates the density of interconnection within the password set. Furthermore, the intriguing structural characteristics provide a possible explanation of the diminishing returns in cracking curves, which is a phenomenon observed in most attacking results over decades [8].

Our final contribution is the model of statistical guessing attack. Based on the proposed model, we focus on the optimal dictionary problem, which aims at cracking a password set with the minimized size dictionary needed. With the knowledge of password distribution, we manage to map the problem of password cracking to the dominating set problem on the graph we visualized and give a theoretical upper bound of the success rate an attacker could ever possibly achieve. Meanwhile, we also demonstrate that the optimal dictionary problem is equivalent to one of the classic NP-complete problems, the minimum dominating set problem, and the complexity for an attacker to find an optimal dictionary is therefore NP-complete.

2 Visualization of the Empirical Password Sets

2.1 Previous Password Set Analyzing Metrics

Characteristics Description. In most cases, the way of presenting the password sets is a list of the characteristics information of the passwords. For instance, many works on password data sets mentioned the top 10 (or higher) most popular passwords of the data involved. Some of descriptions are linguistic classification, in which passwords are classified into different categories such

[1] These data sets were disclosed after a series of serve leakages and were collected subsequently. Each one of the data sets has been mentioned at least once in previous literature. For instance, hotmail in [4], 12306 in [5], phpbb in [6], yahoo in [7]. Details are omitted to conserve space.

as words, places, names, movie lines, email, phone number, home address, etc. Others may focus on common attributes of passwords, like password frequency, length, character composition including but not limited to the occurrence number of digits, lower or upper case letters, special characters and so on. Relevant examples could be found in [9–12] and many others. The major breakthrough comes with the probabilistic password cracking models, including Markov modeling techniques from natural language processing by Narayanan and Shmatikov [13] in 2005 and, later in 2009, the Probabilistic Context-Free Grammars model by Weir et al. [14]. The statistical guessing model is a great leap for password cracking.

Word Cloud of Password Sets. Word cloud is another option when visualizing words. According to the homepage introduction of Wordle[2], which is an online word cloud service provider, the word clouds generated from original text give greater prominence to words that have higher frequency in the source text. Note that the fonts, layouts and color schemes can be tweaked by the users.

Fig. 1. The word cloud of 12306's top 100 mostly used passwords. (Color figure online)

In [15], Wordle was set up to reveal features in the password set of Rockyou, such as the mixed numeric and text dates. Figure 1 is a simple word cloud of the top 100 mostly used passwords in 12306's data set[3].

This method gives more straightforward and obvious information about the password set than the characteristics descriptions. Through the contrast in size, the more important password distinguishes themselves from the ones that weight less. The variations in color also make the visualized data more friendly than a simple list of numbers. Furthermore, some patterns and features of the data stand out easily with the help of word cloud. For example, sequences like "123456", "qwer" (which is a sequence of keys on a standard keyboard), "123" and "woaini" appear frequently in the given data set.

[2] http://www.wordle.net/.

[3] The 12306's data set is one of the data sets used in this paper. Refer to the subsequent sections for more details about the data sets.

2.2 The Definition of Distance Between Passwords

Since we are trying to figure out the relations between passwords, the first thing is to define the relationship of two passwords. In the literature, there seems to be no general definition of the similarity or dissimilarity between two passwords. Before we make decisions in real life, we usually estimate the pros and cons. Likewise, when we try to compare passwords, we measure the similarity or dissimilarity. Hence, what is the difference of two passwords? How to measure the degree of the dissimilarity? Passwords, as we know, are strings of letters, numbers and special characters. The natural choice is, therefore, the way we measure the similarity or dissimilarity of two strings. In this study, we choose edit distance for the measurement of dissimilarity between passwords.

Edit distance is a way to quantify the dissimilarity of two strings (e.g., words) by counting the minimum number of operations required to transform one string into the other. We use one of the most common and well-known variants called Levenshtein distance, which was named after Levenshtein [16]. Levenshtein distance could also simply be referred to as "edit distance", even though several variants exist [17].

The widespread usage of edit distance is a plus, not to mention the corresponding efficient algorithms for utilization. The computing of the edit distance between passwords is based on an improved version of dynamic programming algorithm, which is commonly credited to Wagner and Fischer [18] and has approximately linear time complexity. The computing efficiency is a non-negligible factor to take into account when processing the data, especially when the quantity of the data accumulates to 6 or higher in order of magnitude.

Moreover, edit distance was chosen for the measurement of dissimilarity between passwords because its definition is in accordance with the standard practice of mangling in dictionary based password guessing. The significance of mangling rules has been highlighted and verified by the famous password cracking tool *John the Ripper* and many experts [2,6,8,11,12,14,19] in the field. The aim of our work is to broaden the knowledge of organization and spatial structure of password sets. As shown in the following sections, the visualization of password networks is based on edit distance between passwords. Thus the networks are in some sense the reflection of connections between passwords when they are under attack.

2.3 The Method of Visualization

The procedure to build the graph of a given data set is as follows:

i Each unique password is represented by a single node, also known as a vertex, in the graph;
ii Add an edge between two nodes if the distance $D(i,j)$ between two corresponding passwords is less than a threshold;
iii Repeat step (ii) until every pair of two passwords in the data set has been compared;
iv Reorganize the graph and output the layout of the graph.

The threshold of distance between passwords is on the basis of practical metric and the computing capacity available when dictionary based cracking happens. We choose the threshold of 1, 2, 3 in this paper on account of the fact that the computing complexity becomes unacceptable when the edit distance is larger than 3. Note that the computing complexity we are addressing here is not the complexity of computing the edit distance between two passwords, but the computing complexity when an attacker attempt to crack as many accounts as possible within distance less than the threshold. Though the number of nodes is fixed for a given data set, which is equal to the number of unique passwords, the larger threshold means more edges and a graph with higher density.

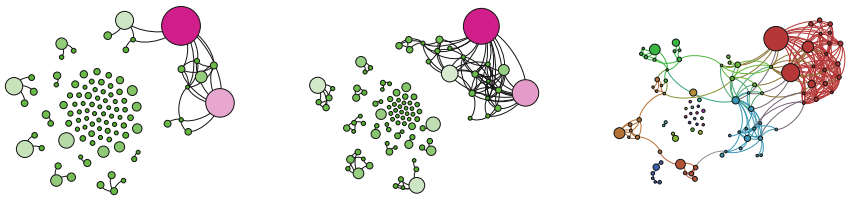
Meanwhile, for sake of space complexity, we compute the distance between every two passwords and store them in form of adjacent table, instead of adjacent matrix.

2.4 A Simple Example of Our Visualization

To make the procedures of our visualization clear and easy to follow, again, we take the top 100 mostly used passwords of the 12306's data as an example. Table 1 is the source data of the passwords. The password number in the table is usually referred to as the frequency of a password, i.e., the number of the same password occurs in the data set. For instance, there are 392 users of website 12306 use "123456" as their passwords and 165 users choose "123456a".

The adjacent table is taken as input for Gephi (an open-source network analysis and visualization software [20]) and the output is the graph of the network within the distance of 1, 2 and 3 separately. The visualization of 12306's top 100 mostly used passwords is shown in Fig. 2a, b, and c. Figure 2a is the graph of the top 100 mostly used passwords within edit distance 1 in 12306 while Fig. 2b and c are the graphs within distance 2 and 3 separately.

Although a graph within edit distance 3 or 2 is obviously much better connected than a graph within edit distance 1, we stop at distance 3 because of computing complexity. The computing complexity grows exponentially when the



(a) edit distance 1. (b) edit distance 2. (c) edit distance 3.

Fig. 2. The graph of 12306's top 100 mostly used passwords within edit distance 1, 2 and 3. (Color figure online)

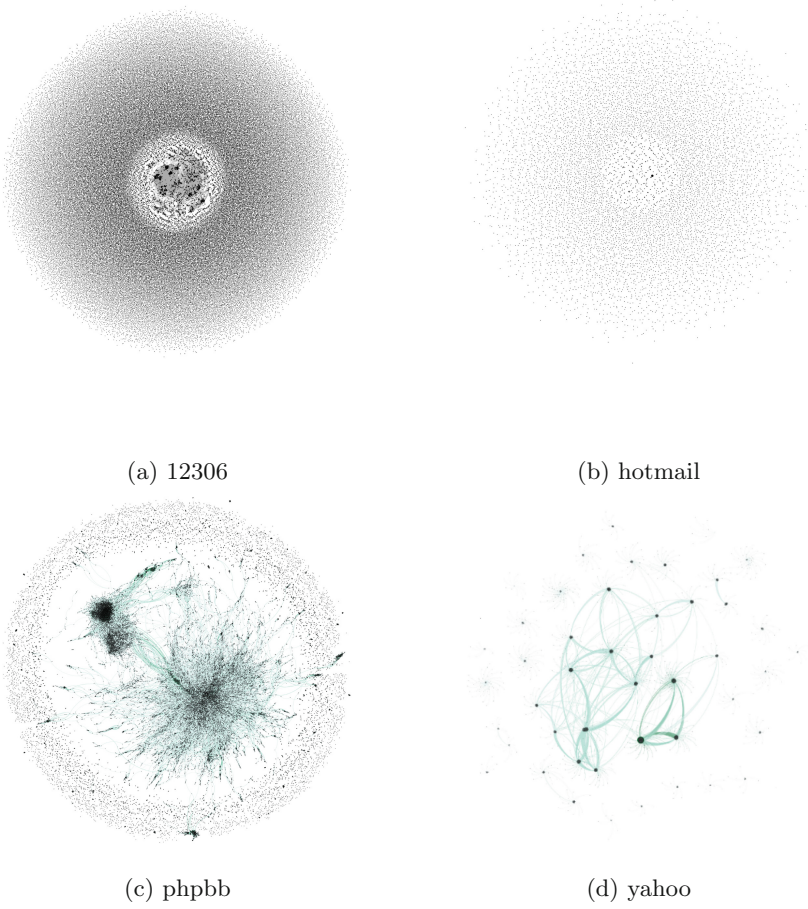
Table 1. 12306's top 100 mostly used passwords and the corresponding frequency

Password	Password number	Password	Password number
123456	392	a123456	281
123456a	165	5201314	161
111111	157	woaini1314	136
qq123456	98	123123	98
000000	97	1qaz2wsx	93
1q2w3e4r	83	qwe123	80
7758521	76	123qwe	68
a123123	63	woaini520	56
123456aa	55	100200	52
1314520	52	woaini	51
woaini123	50	123321	50
q123456	49	123456789	49
123456789a	48	5211314	48
asd123	48	a123456789	48
z123456	47	asd123456	47
a5201314	45	zhang123	42
aa123456	41	123123a	40
aptx4869	38	1q2w3e4r5t	37
1qazxsw2	37	5201314a	36
1q2w3e	35	aini1314	35
woaini521	34	31415926	34
q1w2e3r4	34	123456qq	34
1234qwer	33	520520	33
a111111	33	110110	29
123456abc	29	111111a	29
7758258	28	w123456	28
abc123	28	159753	26
iloveyou	26	qwer1234	25
a000000	25	123654	24
123qweasd	24	zxc123	24
qq123123	23	123456q	23
abc123456	23	qq5201314	22
12345678	22	000000a	21
456852	21	1314521	20
666666	19	asdasd	19
as123456	19	112233	19
521521	19	zxc123456	19
q1w2e3	18	abcd1234	18
aaa123	18	11111111	17
aaaaaa	17	qazwsx123	17
qaz123	17	123000	17
12qwaszx	17	a123321	17
caonima123	16	asdasd123	16
1123581321	16	110120	16
584520	16	zxcvbnm123	16
753951	16	159357	16
nihao123	16	5845201314	16
wang123	16	love1314	16
s123456	16	147258	16
hao123	15	123456asd	15

distance expands. Actually, it is nearly impossible to reach a full estimation of distance 4, according to our result. It is worth mentioning that we also use the variance of size and color of vertex to deliver a better view. In Gephi, the size of vertex is set to be directly proportional to the frequency of a password. In other words, the size of vertex grows when the frequency of a password increases.

From the example of the top 100 mostly used passwords of the 12306, we expose the evolution of the password network within the distance 1, 2, and 3 and visualize the spatial structure of an empirical password set.

As observed in the graph, some nodes in the network are adjacent to a large number of nodes while some other nodes have only a few edges. In particular, a portion of nodes in the graph are isolated. In other words, they are not connected to anyone.



(a) 12306 (b) hotmail

(c) phpbb (d) yahoo

Fig. 3. The visualized graph of 12306, hotmail, phpbb, yahoo's data sets within edit distance 1.

To further analyze the structure of the graph, we take the community and clustering method to separate the network apart and give a more clear vision of the structure of the data set. The network can be partitioned into different communities, depending on their interconnection. The implementation of community detection in Gephi is based on [21]. Different communities are represented in distinct colors, ranging from dark red to light green. Internal nodes in each community (or group) are linked more closely, which means they have more edges among them, while nodes between the communities contact sparsely. To put it another way, there are less edges between communities. Again, we take the top 100 mostly used passwords of 12306 as the example in Fig. 2c. To our surprise, like the social network of human beings, passwords have their own community and social network. As shown in Fig. 2c, the nodes in different "community" are displayed in different colors.

Figure 3a, b, c, d are the graphs of 12306, hotmail, phpbb and yahoo's password sets within edit distance 1 separately. As shown in the graphs above, the distribution of passwords tends to form communities and clusters. To put it another way, some passwords are closer to other passwords and the whole data set is split into different parts. Table 2 gives the number of nodes and edges in the graph.

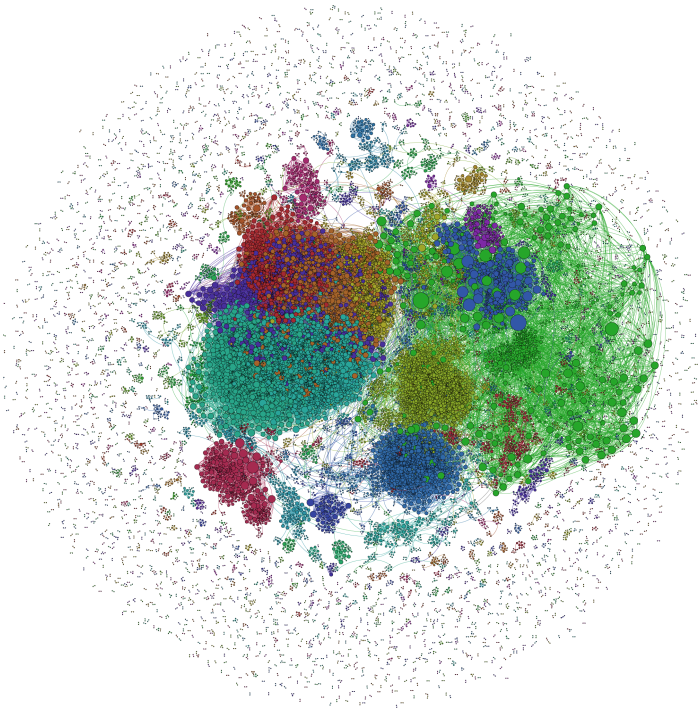


Fig. 4. The visualized graph of 12306's password set within edit distance 3.

To make our observation convincing and solid, we further visualize other data sets avail. Figure 4 is the graph of the full 12306's data set within edit distance 3 after clustering. It is obvious that the drifting isolated nodes is a single community when being analyzed.

Due to a limited number of pages allowed, we only present part of the graphs. Full coverage of graphs on the four data sets ranging from distance 1 to 3 will be available on arXiv[4] with the same paper name and the author's github repository[5].

3 Statistical Analysis on the Visualized Password Networks

The study of networks originates in the ancient graph theory and has become a crucial area in both theoretical research and empirical applications. The electric power grid, the WWW [22] and the pattern of air traffic between airports are early examples of networks in real life. We make friends with others and our friends have friends of their own, so the social network is generated. The boom of social networks in the last decades has made a big step forward in the understanding of social science, as well as the networks of movie actors and scientific collaboration.

Networks are everywhere. As far as we are concerned, the networks that have been studied so far are, to a certain extent, public. The initial motivation of the network is to share or transit information, goods and sorts of data, from one to the other. As the key to access resources or accounts, password, however, is meant to be private in the first place. Unlike the components like people, airports, routers on the Internet that consist of various networks, passwords are chosen independently and are supposed to be personal and private. Unfortunately, it turns out that the passwords generate networks that we have never imagine and that pose inevitable threats for numerous accounts and organizations.

3.1 Statistical Characteristics of the Data

Although not every one of the graphs is displayed in Sect. 2, we conduct a thorough investigation into every result of our visualization. Table 2 is a brief overview of the number of nodes and edges in the graphs of password data sets with the corresponding edit distance ranging from 1 to 3.

As shown in Table 2, the quantity of nodes and edges in the graphs varied from one to the other. For the graph within different distance threshssold, the deviation of the number of edges could be up to 1 or 2 orders of magnitude.

[4] https://arxiv.org/.
[5] https://github.com/googlr/.

Table 2. The number of nodes and edges in the graphs of password sets within edit distance 1, 2 and 3.

Password set	Number of nodes	Number of edges within distance 1	Number of edges within distance 2	Number of edges within distance 3
Hotmail	8,930	742	6,107	45,896
12306	117,808	51,299	676,011	5,311,460
Phpbb	184,341	81,220	1,206,322	13,849,678
Yahoo	342,510	144,209	1,477,190	13,691,942

3.2 Hypothesis of the Degree Distribution in the Networks

The distribution with which passwords are chosen has been an intriguing topic for the researchers in the field. The reason is simple: with a sound knowledge of the distribution of human-chosen passwords, we could utilize the statistical techniques to get a better performance in password cracking, like the PCFG or Markov models. In fact, numerous previous works have made such attempts in revealing features and patterns in password creation and distribution. Malone and Maher [4] claimed that Zipf's law is a good candidate to describle the frequency distribution of password choices, which was later endorsed by Wang in [12]. Now that we have obtained the structure of password data sets, the structural characteristics were further explored in the remaining sections.

Given that the structure of the data sets takes the form of a network and our focus is the interconnection of nodes, the degree of nodes, which is the number of nodes adjacent to the node, incorporates more substantial information. The degree distributions of the visualized graphs of the data sets are shown in Fig. 5. On a typical log-log axes, Fig. 5a, b, c and d are from data set of hotmail, 12306, phpbb and yahoo respectively. The plots of degree distribution are generated by R [23], which is a free software environment and comprehensive language for statistical computing and graphics, and ggplot2 [24] package.

To make a solid statistical analysis of the data and reduce the deviation brought in by randomness and skewness of sampling, if not mentioned particularly, we choose the networks of data sets within edit distance 3 as the source input of the analysis. Large data sets are normally preferred in statistics, because natural noise of sampling and insufficiency of sample size are considered the major shortage of smaller data sets which lead to inaccurate analysis.

In any statistical analysis, it is non-trivial to fit a certain distribution to given data and to measure the goodness of the fit as well. Multiple aspects of the data, including the domain-specific characteristics, should be taken into consideration in particular circumstances, otherwise the fitting could be inaccurate.

Conventionally, the standard statistical method for fitting a common distribution consists of three basic steps: visualizing, fitting, and evaluating [25]. The result of visualizing step is in Fig. 5. In subsequent parts of this section, the fitting step is in Sect. 3.3 and the evaluating step is in Sect. 3.4.

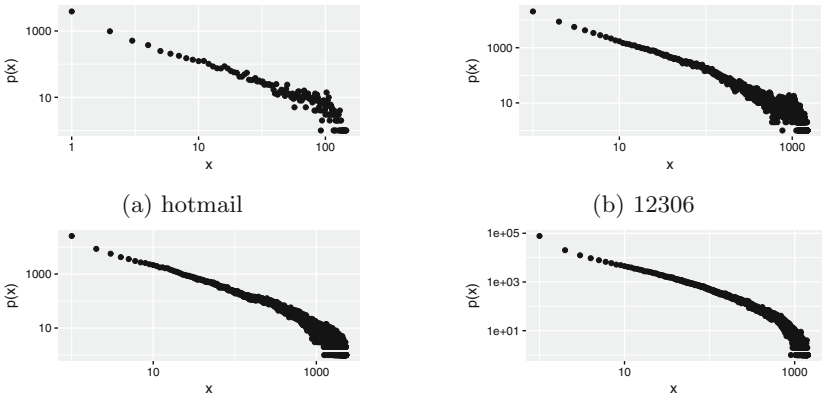


Fig. 5. The degree distributions of the visualized graphs of the data sets within edit distance 3.

3.3 Fitting to Power Law and Estimating of the Scaling Parameter α

Generally speaking, the first problem, when describing empirical data, is to make a hypothesis of the distribution to which the data may follow. This problem is of such vital significance that it directly determines the accuracy of the fitting and, on the other hand, is sometimes quite tricky. As Alstott et al. pointed out in [25], it is possible that, for the given data set, there is more than one distribution fits well, in which case we for some reason choose one as the hypothesis instead of the other. To make things worse, the distribution that fits the data best might occasionally fall into the alternatives and thus slip out of our scope without being noticed, especially when the one we choose could pass the hypothesis test as well. In consequence, with so many candidate distributions to choose from, it usually requires observations from initial tests and experience to make a decision.

From the plots in Fig. 5, each source of the data could be approximated linearly and has a heavy tail, meaning the tail of the data contains a great deal of probability. On the basis of observations and initial tests, we made the assumption that the degree distribution follows the power law. In this section, we will estimate the parameters of the fitting distribution.

The power law distribution, which is sometimes referred as Pareto distribution, is a probability distribution known for its frequent appearance in natural and man-made phenomenon, as well as its complicated properties. The form of power laws is

$$p(x) \propto x^{-\alpha} \tag{1}$$

Mathematically, α, known as the *exponent* or *scaling parameter*, is a constant parameter of the distribution in Eq. 1.

The fitting is performed with the open-source software package *powerlaw* developed and maintained by Alstott et al. [25], which is a Python implementation of the principled statistical framework proposed by Clauset et al. in [26].

Before fitting, we'd like to go over a few crucial points about the fitting techniques. According to Clauset et al. [26], the approach combines maximum-likelihood fitting methods with goodness-of-fit tests based on the Kolmogorov-Smirnov statistic and likelihood ratios. In practice, power law distribution, in most of the cases, only covers a portion of the data in the tail. In other words, the power law behaviour holds merely on a range of given data and the starting point of the range is referred as x_{min}. When fitting a power-law distributional model to data, the approach[6] estimates alpha for each possible x_{min} and select the value that gives the minimum value of Kolmogorov-Smirnov statistic D as the ultimate estimate [25].

The results of estimation are shown in Table 3. In the second and third column of Table 3, est. alpha is the fitted parameter α and *sigma* is its standard error. Note that this procedure gives estimate of fitted parameters, and the validity of the fit will be covered in the next section.

It is often the case that a line is add to show how close the fit is to the data. While, as Clauset et al. [26] pointed out, the conclusion of such observations is more or less objective and should not be trusted, especially when large scale of fluctuation lies in the tail of empirical data.

3.4 Testing the Power-Law Hypothesis

The goodness of fit of hypothesis distribution must be evaluated before coming to the conclusion that the hypothesis distribution is a good description of the data. As a consequence of fluctuations in sampling, the data collected from a non-power-law process might happen to fit the power law distribution, on the other hand, the data drawn directly from a power law distribution could fail the power law hypothesis test. In the view of Clauset et al. [26], it is recommended that one should prefer large statistical samples to reduce the odds of test failure, as which dwindle with increasing sample size.

When it comes to the techniques of goodness-of-fit tests, there are two options: (1) consider the goodness of fit for each distribution individually, in which case a p-value for the hypothesis is generated by using bootstrapping and the Kolmogorov-Smirnov test, and then check the significance level; (2) compare the candidate hypothesis with alternative distributions by using loglikelihood ratios and identify which one is better. Alstott et al. [25] suggest the latter one, the comparative tests.

Table 4 shows the goodness-of-fit between power law and other widespread heavy-tailed distributions. The list of alternative distributions are the exponential distribution, the lognormal distribution, the lognormal-positive distribution, the stretched exponential (Weibull) distribution and the truncated power law (power law with cut-off) distribution. LR is the loglikelihood ratio between the

[6] http://tuvalu.santafe.edu/~aaronc/powerlaws/.

two candidate distributions. This number will be positive if the data is more likely in the first distribution, and negative if the data is more likely in the second distribution. The significance value for the preferred distribution is p.

As usual, the significant level of p is 0.05. From Table 4, the results denoted in **bold** fail ($p < 0.05$) the test and are, therefore, ruled out. From the statistic in the second column, the exponential distribution is not considered to be a proper model. In the third column, there is a fierce competition between the power law distribution and the lognormal distribution. The value of LR is so close to 0 that it is hard to make a trade-off on the sign of LR, which indicates that two distributions are quite close. Or put it in another way, power law is a model that is at least as good as the lognormal model. In the fourth column, the power law model is relatively a better fit than the lognormal-positive model, except a close match for phpbb, in which case power law model is no worse than the lognormal-positive model. In the fifth column of the table, the situation is similar to that of the fourth column and the power law model wins.

Table 3. The estimation result of fitting degree distribution to power law.

Password set	*est.alpha*	*sigma*	x_{min}	D
Hotmail-3	1.8532	0.0909	8.0	0.0536
12306-3	1.7573	0.0240	6.0	0.0311
Phpbb-3	2.1541	0.1542	439.0	0.0492
Yahoo-3	2.2636	0.2694	2307.0	0.0343

When it comes to the last column, the truncated power law, also known as power law with a cut-off model stands out except a close match for yahoo. At this point, it seems that we have made the wrong choice of hypothesis. As Alstott et al. noted in [25], those two-parameter distributions, like the truncated power law and the alternative heavy-tailed distributions, have a natural advantage over the power law, which actually has only one degree of freedom for fitting. However, as long as the model describe the data in a sound and solid way, we say it is a good fit. Actually, we could always find a model with enough parameters to describe the data and eventually trap ourselves into overfitting. Moreover, according to the definition, the truncated power law has the power law's scaling behavior over some range but is truncated by an exponentially bounded tail. It does not make sense to claim that the power law model is a worse fit than the truncated power law model when the latter is a nested distribution of the former. By the way, note that when the indicated conclusions contradict each other, we tend to trust the result on larger sample size.

In conclusion, the power law model is a good fit for the degree distribution of the source data. Meanwhile, the scale-free property is that the degree distribution of complex networks is in accordance with the power-law distribution, and a small number of nodes in the network have a large number of edges. So the

topological distribution of the password sets could be described as a scale-free network, which is naturally true by definition. Results of other data sets agree with the conclusion we made here.

Until now, the conclusion matches with our common sense that popular passwords are widely used and a great number of users tend to use at least similar passwords. From previous works, we have realized that individuals tend to choose same passwords. In this literature, the networks of passwords reveal the fact that users tend to choose similar passwords in a much higher chance. If considered carefully, it does make sense. Though we are individuals and we choose our passwords independently, if we tend to choose same passwords, the odds that we choose passwords that slightly differ from each other is much higher than that we choose the same passwords. Therefore, the security of one single account and the security of the whole system are no longer isolated and, moreover, are connected for the first time. That is generic mechanism from where the network of our passwords begins.

Table 4. Comparison between power law and alternative distributions.

Data set	Exponential	Lognormal	Lognormal-positive	Stretched exponential	Truncated power law
Hotmail	LR = 97.4145	LR = −0.0020	LR = 2.5885	LR = 0.8370	LR = −0.0859
	p = 0.0045	p = 0.6368	p = 0.2035	p = 0.4253	p = 0.6785
12306	LR = 1315.1101	LR = 0.0089	LR = 30.5096	LR = 7.8778	LR = −1.7073
	p = 5.9110e−11	p = 0.7689	**p = 1.3775e−05**	**p = 0.0199**	p = = 0.0646
Phpbb	LR = 21.9523	LR = -0.0836	LR = -0.0459	LR = −0.0213	LR = −0.1468
	p = 0.0730	p = 0.8064	p = 0.9382	p = 0.9755	p = 0.5880
Yahoo	LR = 10.0664	LR = 0.0004	LR = 0.1109	LR = 0.1378	LR = −0.0089
	p = 0.07295	p = 0.8953	p = 0.7108	p = 0.6878	p = 0.8939

4 The Statistical Guessing Model

4.1 A simple model of password guessing

With the knowledge of the entire targeted password set, it is possible to trace the process of a dictionary based password guessing on the graph and to estimate the maximum success ratio.

To estimate the number of potential maximum successful guesses, the concept of neighborhood is introduced. In graph theory, the **neighborhood** of a vertex v, denoted as $N(v)$, is the set of adjacent vertices of G consisting of all vertices adjacent to v in graph $G(V, E)$. Note that the concept of neighborhood we discuss in this paper is the closed neighborhood, in which v itself is included. There is another version of neighborhood is called open neighborhood when v itself is not included [27].

The concept of neighborhood of one vertex can be naturally extended to a set of vertices V_s, which is the union of all the neighborhoods of the vertices in set V_s, meaning that each of the vertices in the original graph is adjacent to at least one member of V_s. Denoted as $N(V_s)$ and we have

$$N(V_s) = \cup_{i=1}^{|V_s|} N(v_i) \tag{2}$$

in which $|V_s|$ is the number of vertices and v_i is the i-th vertex in V_s.

Given a dictionary of n passwords $Dict = \{p_1, p_2, \ldots, p_{n-1}, p_n\}$. The passwords are arranged in decreasing order of frequency, i.e.

$$f(p_1) > f(p_2) > \cdots > f(p_{n-1}) > f(p_n),$$

where $f(p_i)$ is the frequency of the password p_i in the targeted password set **T**.

As shown previously, we could build the graph of any specified password data sets. In the corresponding data set, if the attacker guesses one password right, the vertex for which the compromised password stands is covered by the attacker's dictionary. For each vertex that is directly adjacent to the compromised vertex, the attacker could cover them all within affordable time. More details about this one to one mapping mechanism will be stated afterwards in the optimal dictionary problem.

Then for an attacker with dictionary **Dict**, the maximum set of vertices could be covered in the graph of target **T** is the union of passwords that **Dict** covered and their neighbors in the graph of **T**, which is

$$N(Dict) = \cup_{i=1}^{n} N(p_i). \tag{3}$$

Thus the total number of corresponding maximum successful guesses is $\sum_{p \in N(Dict)} f(p)$. The upper bound of the success ratio using dictionary **Dict** is the accumulation of the frequency of the node and its neighbors. Of course the attacker can start multiple rounds by searching the closure of the compromised data, but the overall time cost could be intolerable.

4.2 The Optimal Dictionary Problem

In conventional password cracking, the size of dictionary has a significant impact on the success rate of the cracking. The primary reason that attackers prefer large dictionary is straightforward: a larger dictionary means the higher probability of covering more passwords in the targeted set. Meanwhile, due to the efficiency of time and space, all results show diminishing returns as the dictionary size swells [8]. The diminishing guessing curves have been observed in almost every previous attempt to crack as more accounts as possible.

Klein [28] made the first attempt to identify the higher efficient subdictionary. J Bonneau define a success rate α when introducing α-guesswork to evaluate the number of guesses of an attacker [2]. And Mónica and Ribeiro [29] discussed the compression ratio in the implementation of Self-Organizing Maps (SOM) model which preserves the topological position of passwords.

Since we have built a graph of the password set, the search for better subdictionary becomes easier. Our goal is to find a subset of strings to cover as more passwords as possible. Considering we are dealing with this problem on a graph, if we paraphrase the problem a little bit, the goal is to find a subset of nodes that all the other nodes in the graph of the target are adjacent to at least one member of this subset. That is exactly the definition of dominating set in graph theory. Given a graph $G = (V, E)$, a dominating set for a graph G is a subset D of V that every vertex in V is either in D or adjacent to at least one member of D. The number of vertices in a smallest dominating set for G, $\gamma(G)$, is known as the domination number. Refer to [30] for more details of the definition.

To be mathematically precise and concise, we proposed the reductions below to show the equivalence of the optimal dictionary problem of password guessing and the minimum dominating set problem.

For any password set $S = \{p_1, p_2, \ldots, p_n\}$, we can construct the graph $G = (V, E)$ within certain distance threshold through the steps in Sect. 2, which mainly involves in generating the edges and takes polynomial time.

Note that there is a one-to-one mapping between the passwords in S and the nodes in G. Let $\hat{D}$ be an instance of the optimal dictionary of S, meaning that $\hat{D}$ is a minimum subset that is able to recover S. In graph G, the set of nodes which represents the elements of $\hat{D}$ is denoted by D. Now consider the situation in G, we have $V \subseteq N(D)$, in which case D is a dominating set of G.

The next step is to validate that D is a minimum dominating set of G. Assume that D is not a minimum dominating set of G, which indicates that either D is not a dominating set of G or D is a dominating set but not the smallest. In the former situation, at least one node, say p_k, neither belongs to D nor is adjacent to any member of D. Backing to the source data set, the password that p_k represent is neither in $\hat{D}$ nor recoverable by $\hat{D}$, which is contradiction to the our proposition that $\hat{D}$ is a dictionary of S. While in the latter situation that D is not the smallest dominating set, suggesting that at least one node p_t could be removed from D and $D^* = \{D - p_t\}$ serves as a smaller dominating set of G. Then if we remove the password that p_t represent from $\hat{D}$, $\hat{D}^* = \{\hat{D} - p_t\}$, which is smaller than $\hat{D}$ and could also recover S, leads to a contradiction that $\hat{D}$ is not optimal. To summarize, D is a minimum dominating set. Likewise, we can generate an optimal dictionary with a given minimum dominating set of G. As a result, given an instance of the optimal dictionary problem, we can construct an instance of the minimum dominating set problem and vice versa.

The complexity of transformations are polynomial time. In other words, the minimum dominating set problem and the optimal dictionary problem are equivalent in terms of computing complexity. The minimum dominating set problem is a well-known NP-hard problem, which is proved by Garey and Johnson in [31]. Hence the minimum size of the dictionary to cover the targeted password set, i.e. its lower bound, equals $\gamma(G)$. Note that this conclusion also applies to other variants of dictionary based cracking techniques, provided that the corresponding method to build the graph is properly redefined.

5 Conclusion

In this paper, we provide a novel presentation of empirical password sets in the form of networks from scratch. The spatial structure of the password sets is discussed for the first time and is considered to be a scale-free network.

The high density of interconnections between passwords provides a candidate explanation of the diminishing returns observed in previous literature. While many users choose the same password in reality, It went unnoticed that more users tend to choose similar passwords. To make things worse, the difference between those passwords is usually negligible against the computing capacity nowadays and even the strong password could not resist when the chain reaction of leakage started.

Furthermore, at the basis of the network graph of password set we proposed, we give the upper bound of the maximum password attacking success rate based on a certain dictionary. Under the assumption of an attacker who has high performance computing resource, we demonstrate the equivalence of the optimal dictionary problem and the dominating set problem in computing complexity. Therefore the optimal dictionary problem is also NP-complete.

Acknowledgement. The authors would like to thank Ping Wang, Tian Liu, Yongzhi Cao, Wenxin Li, Eric Liang, Kaigui Bian, Haibo Cheng, Ding Wang, Gaopeng Jian, Chen Zhu, Xin Huang, Qiancheng Gu, Hang Li, Jun Yang, Junfeng Zhang, Xuqing Liu, Xiangyu Xu, Xiang Yin, Wenying Teng, Meredith Mante, Justin Edwin Marquez, Alex Wilke and Niall Pereira for helpful conversations and the anonymous reviewers for their insightful comments. This work was sponsored by the National Science Foundation of China under grant No. 61371131.

References

1. Schechter, S., Herley, C., Mitzenmacher, M.: Popularity is everything: a new approach to protecting passwords from statistical-guessing attacks. In: Proceedings of the 5th USENIX Conference on Hot Topics in Security, pp. 1–8. USENIX Association (2010)
2. Bonneau, J.: The science of guessing: analyzing an anonymized corpus of 70 million passwords. In: 2012 IEEE Symposium on Security and Privacy, pp. 538–552. IEEE (2012)
3. Morris, R., Thompson, K.: Password security: a case history. Commun. ACM **22**(11), 594–597 (1979)
4. Malone, D., Maher, K.: Investigating the distribution of password choices. In: Proceedings of the 21st International Conference on World Wide Web, pp. 301–310. ACM (2012)
5. Carnavalet, X.D.C.D., Mannan, M.: A large-scale evaluation of high-impact password strength meters. ACM Trans. Inf. Syst. Secur. (TISSEC) **18**(1), 1 (2015)
6. Weir, M., Aggarwal, S., Collins, M., Stern, H.: 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. ACM (2010)

7. Das, A., Bonneau, J., Caesar, M., Borisov, N., Wang, X.: The tangled web of password reuse. In: NDSS, vol. 14, pp. 23–26 (2014)
8. Dell'Amico, M., Michiardi, P., Roudier, Y.: Password strength: an empirical analysis. In: INFOCOM, vol. 10, pp. 983–991 (2010)
9. Mazurek, M.L., Komanduri, S., Vidas, T., Bauer, L., Christin, N., Cranor, L.F., Kelley, P.G., Shay, R., Ur, B.: Measuring password guessability for an entire university. In: Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, pp. 173–186. ACM (2013)
10. Voyiatzis, A.G., Fidas, C.A., Serpanos, D.N., Avouris, N.M.: An empirical study on the web password strength in Greece. In: 2011 15th Panhellenic Conference on Informatics (PCI), pp. 212–216. IEEE (2011)
11. Li, Z., Han, W., Xu, W.: A large-scale empirical analysis of Chinese web passwords. In: USENIX Security Symposium, pp. 559–574 (2014)
12. Wang, D., Cheng, H., Wang, P., Huang, X., Jian, G.: Zipf's law in passwords. IEEE Trans. Inf. Forensics Secur. **12**(11), 2776–2791 (2017)
13. Narayanan, A., Shmatikov, V.: 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 (2005)
14. Weir, M., Aggarwal, S., De Medeiros, B., Glodek, B.: Password cracking using probabilistic context-free grammars. In: 2009 30th IEEE Symposium on Security and Privacy, pp. 391–405. IEEE (2009)
15. Veras, R., Thorpe, J., Collins, C.: Visualizing semantics in passwords: the role of dates. In: Proceedings of the Ninth International Symposium on Visualization for Cyber Security, pp. 88–95. ACM (2012)
16. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. In: Soviet Physics Doklady, vol. 10, p. 707 (1966)
17. Navarro, G.: A guided tour to approximate string matching. ACM Comput. Surv. (CSUR) **33**(1), 31–88 (2001)
18. Wagner, R.A., Fischer, M.J.: The string-to-string correction problem. J. ACM (JACM) **21**(1), 168–173 (1974)
19. Ur, B., Segreti, S.M., Bauer, L., Christin, N., Cranor, L.F., Komanduri, S., Kurilova, D., Mazurek, M.L., Melicher, W., Shay, R.: Measuring real-world accuracies and biases in modeling password guessability. In: USENIX Security Symposium, pp. 463–481 (2015)
20. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An Open Source Software for Exploring and Manipulating Networks (2009)
21. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech: Theory Exp. **2008**(10), P10008 (2008)
22. Barabási, A.-L., Albert, R., Jeong, H.: Scale-free characteristics of random networks: the topology of the world-wide web. Phys. A: Stat. Mech. Appl. **281**(1), 69–77 (2000)
23. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2016)
24. Wickham, H.: ggplot2: Elegant Graphics for Data Analysis. Springer, New York (2009). https://doi.org/10.1007/978-0-387-98141-3
25. Alstott, J., Bullmore, E., Plenz, D.: powerlaw: a python package for analysis of heavy-tailed distributions. PLoS ONE **9**(1), e85777 (2014)
26. Clauset, A., Shalizi, C.R., Newman, M.E.: Power-law distributions in empirical data. SIAM Rev. **51**(4), 661–703 (2009)

27. Hell, P.: Graphs with given neighborhoods i. In: Proc. Colloque, Inter. CNRS, Orsay, pp. 219–223 (1976)
28. Klein, D.V.: Foiling the cracker: a survey of, and improvements to, password security. In: Proceedings of the 2nd USENIX Security Workshop, pp. 5–14 (1990)
29. Mónica, D., Ribeiro, C.: Local password validation using self-organizing maps. In: Kutyłowski, M., Vaidya, J. (eds.) ESORICS 2014. LNCS, vol. 8712, pp. 94–111. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11203-9_6
30. Hedetniemi, S.T., Laskar, R.C.: Bibliography on domination in graphs and some basic definitions of domination parameters. Discret. Math. **86**(1), 257–277 (1990)
31. Garey, M., Johnson, D.: Computers and Intractability-A Guide to NP-Completeness (1979)