Hindawi
Security and Communication Networks
Volume 2018, Article ID 6160125, 12 pages
https://doi.org/10.1155/2018/6160125

Research Article

An Alternative Method for Understanding User-Chosen Passwords

Zhixiong Zheng,[1] **Haibo Cheng**,[2] **Zijian Zhang**,[1] **Yiming Zhao**,[3] **and Ping Wang**[3,4,5]

[1] *School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Shenzhen 518055, China*
[2] *School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China*
[3] *School of Software and Microelectronics, Peking University, Beijing 102600, China*
[4] *National Engineering Research Center for Software Engineering, Peking University, Beijing 100871, China*
[5] *Key Laboratory of High Confidence Software Technologies (PKU), Ministry of Education, Beijing 100871, China*

Correspondence should be addressed to Ping Wang; pwang@pku.edu.cn

Received 30 August 2017; Revised 2 December 2017; Accepted 27 December 2017; Published 28 January 2018

Academic Editor: Qi Jiang

We present in this paper an alternative method for understanding user-chosen passwords. In password research, much attention has been given to increasing the security and usability of individual passwords for common users. Few of them focus on the relationships between passwords; therefore we explore the relationships between passwords: modification-based, similarity-based, and probability-based. By regarding passwords as vertices, we shed light on how to transform a dataset of passwords into a password graph. Subsequently, we introduce some novel notions from graph theory and report on a number of inner properties of passwords from the perspective of graph. With the assistance of Python Graph-tool, we are able to visualize our password graph to deliver an intuitive grasp of user-chosen passwords. Five real-world password datasets are used in our experiments to fulfill our thorough experiments. We discover that (1) some passwords in a dataset are tightly connected with each other; (2) they have the tendency to gather together as a cluster like they are in a social network; (3) password graph has logarithmic distribution for its degrees. Top clusters in password graph could be exploited to obtain the effective mangling rules for cracking passwords. Also, password graph can be utilized for a new kind of password strength meter.

1. Introduction

The invention of computers ushered in a new era of digital lives, and text-based passwords have almost dominated human-computer authentication since then. Passwords continue to prevail on the web as the primary method for user authentication despite consensus among researchers that people deserve something more secure and user-friendly. Many researches have focused more on the specific point of the problem of user authentication involving no user interactions, which is literally the weakest point of the password authentication. There is all the time alternative authentication mechanisms aiming to outright replace password-based authentication method proposed by a bunch of researchers: graphical passwords [1], authentication based on biometric [2], authentication based on user behaviors [3], single sign-on system, and so forth. However, the growing password-based authentication adoptions [4] irrespective of their well-known security and usability drawbacks [5–7] reveal the fact that each of the alternatives to password-based authentication has its own shortcomings as compared to passwords [8]. The inertia of user habits, uncertain transition costs, and constant improvements in passwords lead to the result that incumbent passwords will continue as a stubborn signal for identity authentication in the foreseeable future, where the goal is not impregnable defense but the balance between usability and security [8].

Surveys are conducted by many researches [9–11] to reveal user real-world behaviors in managing passwords. The conflict between users' limited memory and growing number of passwords they need to organize is the main challenge users are confronted with presently. Users reuse passwords or embed mnemonic information into password [12] to alleviate their burden of memorizing passwords. This implies the unpleasant fact that users select usability over security unconsciously. Security experts have suggested the use of password vaults (managers/wallets) to assist password-based authentication with which users need merely to remember one master password to encrypt all their online account passwords in the vault [13, 14].

Before 2009, password researchers proposed some heuristic methods to study the password-based authentication whose main purpose is to reveal the weaknesses of passwords and outright replace passwords in the end [5, 15]. After 2009, large-scale password datasets have been breached widely from hackers' attacking or intruders' intrusion, which are afterwards publicly available, for example, the leak of 32 M passwords from the gaming website RockYou in 2009 is currently the biggest corpus available to the public. In the literature of password, breaching of large-scale datasets of passwords have guided the studies of passwords to a more scientific and rigorous method. Password corpora have typically been used to analyze the distribution of passwords [16], names used in passwords and character distribution [9], or the priority among priorities: used as learning set to train probabilistic models such as PCFG-based [17], Markov-based [18], and NLP-based (Natural Language Processing) [19] password cracking algorithms in order to simulate adversarial password cracking process, leading to sophisticated password dataset strength evaluation methods [20]. The probabilistic password cracking models are afterwards modified elaborately by researchers to evaluate single password strength.

1.1. Motivations. The analysis of the password security can date back at least to Morris and Thompsons 1979 seminal analysis of 3,000 passwords [21]. The analyzing methods they employed can be classified into two categories: password cracking and semantic evaluation. However, these two methodologies focus solely on the individual passwords and neglect implicitly the relationships between passwords. The neglected relationships on the contrary is remarkably important properties of password dataset that could facilitate the password cracking and semantic evaluation. Due to the intrinsically incomplete evaluation of traditional semantic and cracking methodologies, we advocate a new alternative method understanding user-chosen passwords.

In mathematics, graph is mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices, nodes, or points which are connected by edges that can either be directed or undirected. Graph-theoretic methods, in various forms, have been proven particularly useful in many fields such as linguistics, chemistry, physics, and sociology. While there is still no application of graph theory in the literature of passwords research, therefore, we are going to apply graph theory to password dataset to dig deeper into the inner properties of passwords to assess and compare the inherent security behaviors of users. We call this analysis method *relationship evaluation* as an extension to semantic evaluation. Our paper provides an alternative view of password relationships through password graph, we also provide a visualizing method for the generated password to observe and study it intuitively.

1.2. Contributions. In this work, we make the following key contributions:

(i) *An alternative view of password relationships*: we explore the relationships between passwords: modification-based, similarity-based, and probability-based. The modification-based relationship builds on the observation that a user usually modifies an existing password to retrieve a new one. Passwords are basically strings; thus we borrow the idea from string similarity to develop the password similarity-based relationship. The probability-based relationship is the idea derived from password distribution where each password has probability associated with it.

(ii) *Visualizing of password graph*: by regarding passwords as vertices and leveraging one of the relationships we explored, we are able to transform a password dataset to a graph and we call it *password graph*. With the assistance of Python Graph-tool, we visualize our generated password graph; our visualization method can intuitively convey deeper characteristics of password dataset which lay otherwise under the hood and remain undiscovered.

(iii) *Some insights from graph theory*: the resulting password graph provides us a fresh new perspective of password dataset. We will revisit some key terminology in graph theory to find out what our new password graph bring about.

1.3. Organizations. In Section 2, we review prior research works on password cracking and semantic evaluations. Section 3 provides some preliminaries. Section 4 details our exploration of password relationships in dataset of password. Section 5 elaborates on our construction of password graph. Section 6 provides some key insights from graph theory to our password graph. Finally, Section 7 concludes our paper.

2. Related Work

In this section, we briefly review prior pivotal research on password cracking and semantic evaluations to assist our follow-up discussions and explorations.

In 1979, Morris and Thompson analyzed a database of 3,000 passwords and reported some basic statistics: 71.12% passwords of their sample of passwords were 6 characters or fewer and 86% fell into one of the dictionaries, name lists, and the like. In 1990, Klein [22] collected/etc/passwd files in which passwords were in hash format from his friends and acquaintances in United States and Great Britain. 21% passwords were cracked in the first week, total approximately 25% of the passwords had been guessed, and 51.70% of the cracked passwords are not longer than 6 characters. Dedicated cracking software tools like John the Ripper [23] and hashcat [24] have appeared since and are armed with numerous cracking modes (e.g., brute-force attack, and dictionary attack)

Mangling rules in dictionary attack mode continue to evolve beyond heuristic rules: Weir et al. [17] built a machine learning technique based on context-free grammar to automatically derive mangling rules from a large training set of cleartext passwords. Houshmand and Aggrawal [18] derived Markov-based password cracking algorithm from Markov-Chain that representatively originated PageRank algorithm. Originally, Markov-based algorithm is not a probabilistic model; Ma et al. [25] investigated password characteristics about length and the structure of 6 datasets, 3 of which were from Chinese websites and improved it by using different normalization and smoothing methods. They found that when done correctly, Markov-based cracking model performed better than PCFG-based password cracking model. In 2012, Veras et al. [26] had done the work quite similar to us; they examined 32 M RockYou dataset by employing visualization techniques. They observed that 15.26% of passwords contained sequences of 5–8 consecutive digits, 38% of which could be further classified as dates, but their research mainly focused on password patterns which are different from ours.

The guessing resistance of a single user-chosen password was previously estimated by entropy, with reference to Claude Shannon's famous measure H_1. Borrowing the idea of Shannon entropy, a variation of Shannon entropy was proposed in NIST Electronic Authentication Guideline. It calculated the password entropy mainly based on the length of passwords and added partial points if some special heuristic checks were passed to make it more secure. Unfortunately, Shannon entropy and its variations characterize the strength of a distribution; for an attacker who wants just to crack a certain proportion of all passwords, Shannon entropy has no direct correlation to the guessing difficulty. Ad hoc metrics (password strength meter) had already been demonstrated far from accurate by Weir et al. [27]; they advocated that the cracking-based password strength meter was more compelling. Markov-based PSM [28] and PCFG-based PSM [18] were proposed subsequently based on Markov model and PCFG model. Wang et al. [12] created a novel PSM on a solid foundation where user usually reuses one of his/her passwords rather than creating a new one. By using two training sets (one as base dictionary and the other as rule learning dictionary), their PSM was able to derive empirically users' mangling rules on passwords and thus more accurate.

In 2012, Bonneau conducted a large-scale analysis of 70 million Yahoo private passwords and proposed that a more direct password strength metric was guesswork (G) [29]. Yet in 2014, Li et al. [30] came to a conclusion that Bonneau's 70 M passwords were not representative enough for all users, especially Chinese users who are not familiar with English. Chinese users prefer digits than letters. They also showed that Chinese users inclined to insert Pinyins and dates into their passwords.

Semantic patterns including personal information (e.g., birth dates, personal names, and nicknames) are prevalently embedded in user-chosen passwords [31]. Additionally, a few basic data characteristics like average length, length distribution, and types of characters used were typically reported. Further, the structure patterns were also studied by some researchers [9], passwords containing digits constituted more than 50% Chinese web passwords while this value of English counterpart was only 11.30%, reinforcing the hypothesis that user-generated passwords were greatly influenced by their native languages. More systematic methodologies had been proposed by Shay et al. [32] for creating a new password policy. One inconspicuous thing that semantic evaluations fail to attain is the relationships between passwords in the same dataset. Therefore, we for the first time augment this kind of evaluation method by introducing password relationship and graph theory into password evaluations. We hope this evaluation method helps to study the password evaluations thoroughly.

In 2010, Zhang et al. [33] found that modifications to one's old passwords tend to be predictable, and they utilized this observation to facilitate password cracking. Their work is actually password reuse by a certain user; it is independent from other users. Our work focuses on password relationships between different users and classifies them into three classifications.

As far as we know, the work by Guo et al. [34] may be the closest one to what we have done in this work. They visualized several password datasets including Yahoo!, PhpBB, MySpace, Honeynet, hotmail, and 12306. Their discovery provides an explanation of the attacking curve that has long been observed in decades. We find one of their conclusions is wrong: degrees of passwords follow logarithmic law not power law as they claimed; we will detail this in Section 6. Overall, though our works are a bit similar, there are still very critical differences between our work and their work: (a) we explore the relationships between passwords while they do not the relationships between passwords; (b) we explore the effects of different thresholds t while they only experimented with threshold 3; (c) our insights obtained from visualized graph is essentially distinct from their work.

3. Preliminaries

In this section, we explicate formal definitions of graph and linear regression and then introduce the metric for evaluating how well the regression line approximates the real data points. Finally, we provide some basic information on our experimental datasets.

Mathematical Notation. We denote a password dataset with a calligraphic letter $\mathcal{S}$, a password after duplicates removing with another calligraphic letter $\mathcal{P}$. Let N denote the total number of passwords in $\mathcal{P}$.

3.1. Graph

Formal Definition of Graph. A graph G can be defined as a pair $\{V, E\}$, where V is a set of vertices and E is a set of edges between the vertices, $E \subseteq \{(u, v) \mid u, v \in V\}$:

$$G = \{V, E\}. \tag{1}$$

Generally, graphs can be classified into two types: (a) undirected graph, the adjacency relation defined by the edges is symmetric. (b) Directed graph is a graph in which edges have orientations. A simple graph is an undirected graph in which both parallel edges and loops are disallowed while multigraph otherwise allows them.

3.2. Linear Regression. In statistics, linear regression is an approach for modeling the correlation between two variables (one as a scalar dependent variable denoted by y and the other as explanatory variable denoted by x) by fitting a linear equation to the experimental data. The most common method for linear regression is least-squares. Usually, in linear regression, given the value of explanatory variable x, the value of dependent variable y is an affine function of x: $y = k \cdot x + b$, the slope of the line is k, and b is the intercept.

In linear regression, the statistical measure of how well the regression line approximates the real experimental data points is often calculated and compared through the coefficient of determination. People usually denote the coefficient of determination by R^2, which has the range from 0 to 1, the closer to 1 the better. Therefore, a R^2 value of 1 indicates that all experimental data points are perfectly positioned on the regression line; R^2 of 0 indicates the contrary result.

3.3. Spearman's Coefficient. In statistic, Spearman's coefficient is a nonparametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. The Spearman coefficient ρ is defined as the Pearson correlation coefficient between the two ranked vectors X, Y:

$$\rho = \frac{\operatorname{cov}(rg_X, rg_Y)}{\rho_{rg_X} \rho_{rg_Y}}, \tag{2}$$

where $\operatorname{cov}(rg_X, rg_Y)$ is the covariance of the rank variables; ρ_{rg_X} and ρ_{rg_Y} are the standard deviations of the rank variables. By definition, $\rho \in [-1, 1]$, where 0 indicates independence between the vectors. A perfect Spearman correlation of 1 or −1 happens when the agreement between the vectors is a monotone function.

3.4. Password Guess Number. The password guess number characterizes the time complexity required for a password cracking algorithm (PCFG-based or Markov-based) to recover a password. This is generally achieved by measuring the guess number required to crack the password. Dell'Amico and Filippone [20] detail a Monte Carlo sampling method that converts a password pw probability as computed by PCFG or Markov model into an estimate of cracker's guess number:

$$G(\mathrm{pw}) = \frac{1}{k}\sum_{i=1}^{k} \frac{\delta(p_i)}{\Pr(p_i)} + 1$$
$$\delta(p_i) = 1, \quad \text{if } \Pr(p_i) > \Pr(\mathrm{pw}), \text{ else } 0, \tag{3}$$

where k is the sample size and p_i is the password draw randomly from the corresponding distribution.

Table 1: Basic information about our 5 password datasets.

Dataset	Web service	When leaked	Total PWs
MySpace	Social networking	Oct. 2006	49,623
PhpBB	Programmer forum	Jan. 2009	255,373
Rootkit	Hacker forum	Feb. 2011	69,324
Yahoo	Web portal	July. 2012	442,834
12306	Train ticketing	Dec. 2014	129,303

3.5. Datasets. For completeness, we give a brief description of the five datasets used in our experiments (see Table 1). They were breached either by hackers or by intruders and later disclosed publicly on the Internet; some of them have already been used in password cracking models [17, 18].

MySpace was originally published in October 2006, which was obtained by a phishing attack and thus might contain weak as well as fake passwords. Our collection of MySpace list has total 49,623 passwords, but a recent report said that there were actually over 360 million accounts involved. Each record of MySpace list contained an email address, a password, and, in some cases, a second password. Some accounts had multiple passwords; there were in fact over 427 million total passwords available for sale [35]. Rootkit's entire MySQL database backup was released by anonymous group using HBGary's CEO Twitter account; it initially contained 71,228 passwords cryptographically scrambled using the MD5 hashing algorithm; we managed eventually to recover 97.33% of them through trawling attacks. In 2012, nearly 443 K email addresses and passwords for a Yahoo site were exposed after a server breach; hackers gotten into Yahoo's Contributor Network database by using a rudimentary attack called SQL injection. The PhpBB dataset includes about 250 K passwords leaked from Phpbb.com in January 2009; the hacker even described the whole attack in detail on his blog. At the end of year 2014, a password dataset from a Chinese train ticketing website 12306 was leaked to the public by anonymous attackers; it includes nearly 130 K passwords. User information registered on the ticket booking website included real names, ID card numbers and phone, and email contacts and may also include information about family members and friends, posing a serious threat to their information privacy.

4. Password Relationships

We divide our findings of password relationships into three categories: modification-based, similarity-based, and probability-based. Elaborations on these three categories are detailed in the following three subsections.

4.1. Modification-Based. Naturally, the relationship between passwords is actually in some extent tied to users' modifications to passwords. So it could be well imitated by edit distance featuring a set of operations on passwords. Edit distance is a way of quantifying how dissimilar two passwords are to one another by counting the minimum number of operations required to transform one password into the other. However, there are various definitions of edit distance featuring different sets of editing operations: *Levenshtein distance* [36] (hereafter: LD) allows three operations: removal, insertion, and substitution of a character in the password; *longest common subsequence distance*'s permitted operations is a subset of LDs: insertion and deletion; *hamming distance* takes effect only on passwords of same length; that is, hamming distance does not allow insertion or removal. Additional primitive operations like transposition are used by Jaro-Winkler distance based on the observation that a common mistake when people typing is the transposition of two adjacent characters in a string.

The LD between two passwords is formally defined as the minimum number of single-character edits one must perform to change one password into the other. Mathematically, the LD between two passwords a and b can be given by $\mathrm{lev}_{a,b}(|a|,|b|)$:

$$\mathrm{lev}_{a,b}(i,j) = \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+f(a_i,b_j), \end{cases} \quad (4)$$

where $|a|$ and $|b|$ denote the length of passwords a and b, respectively. $f(a_i,b_j)$ is an indicator function that is equal to 0 when $a_i = b_j$ and equal to 1 otherwise. The calculation of LD is actually a dynamic optimization; the time complexity of this algorithm is $O(|a| \cdot |b|)$.

4.2. Similarity-Based. Passwords are essentially strings, so the measures of depiction of string similarity provide us substantial methods to evaluate the similarity (relationship) between passwords. Dice's coefficient is a statistic used for comparing the similarity between two sets and can be used to evaluate the similarity between two passwords: assume X and Y are the set (set element is unique) of characters that passwords a and b include, respectively. Dice's coefficient of a, b is defined as follows:

$$\mathrm{QS} = \frac{2|X \cap Y|}{|X| + |Y|}, \quad (5)$$

where $|X|$ and $|Y|$ are the numbers of elements of the two sets. QS is the quotient of similarity and ranges between 0 and 1.

Jaccard index measures similarity between two finite sets; it is defined as the size of the intersection divided by the size of the union of the sets X and Y (X and Y have the same meaning as described above):

$$J(X,Y) = \frac{|X \cap Y|}{|X \cup Y|} = \frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}. \quad (6)$$

If X and Y are both empty, define $J(X,Y) = 1$, so $J(X,Y)$ is also the quotient of similarity and ranges between 0 and 1.

4.3. Probability-Based. From PCFG-based or Markov-based password cracking model, we can assign probability to every password in a dataset. In principle, two passwords are similar in a sense (same probability that will be picked by a human user) if their corresponding probabilities are close enough, the closer the more similar. One of the workable measures of similarity between passwords a and b may be

$$S(a,b) = \frac{(\Pr(a) - \Pr(b))^2}{(\Pr(a) + \Pr(b))^2}, \quad (7)$$

when a and b have the same probability, their value of $S(a,b)$ equals 0; an extreme situation happens when there is only one password in the distribution; the probability of that password is 1; all other passwords' probability is 0. The value of $S(a,b)$ (where $\Pr(a) = 1, \Pr(b) = 0$) equals 1.

In summary, the relationships between passwords we describe above are only a proportion of the countless relationships of objective existence. For example, the edit distance between passwords is not a measure of similarity but the number of editings. One can of course divide it by the longer length of two passwords, which makes LD remarkably relative LD:

$$\mathrm{lev}_{\mathrm{rel}_{a,b}(|a|,|b|)} = \frac{\mathrm{lev}_{a,b}(|a|,|b|)}{\max\{|a|,|b|\}}. \quad (8)$$

Ultimately, the value of relative LD for two chosen passwords falls into interval $[0,1]$ in which left boundary 0 means exactly the same and right boundary 1 means altogether different.

4.4. Rationale. The practical and accurate measure of depiction of password relationships depends on the particular application scenario. There is no the best depiction method for generating password graph but the most appropriate according to the concrete scenario. For example, if we want to study the modification habits of users, we may choose modification-based classification; but if we only want to learn the password cracking property, we may choose probability-based classification. We attempt to expand our work by utilizing modification-based relationship between passwords (other relationships we have explored above are still worth deeper research and we leave it for future work). Prior surveys have already provided us with plenty of information on how users modify their passwords: adding digits or symbols at the beginning/end of password, capitalizing a letter, Leet transformation, and so forth [9, 12, 32, 37]. Taking the permitted operations for each type of edit distance into consideration synthetically, we could infer that LD is currently the most appropriate measure of password relationship among all types of edit distance, which actually models real-world users' mangling behaviors (insertion, deletion, and substitution are popular while transposition is not). In Section 5, we are going to utilize LD to formally define the edges in a graph.

5. Construction of Password Graph

Generally speaking, any entity that can be described by objects with relationships between them can be regarded as a graph. Since our principal purpose is to characterize the relationship between passwords, We shed light on this notion by (1) regarding each unique password in dataset as

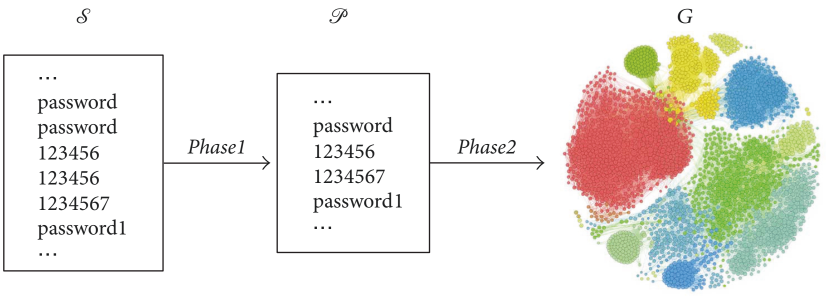


Figure 1: Construction phases.

a vertex: $V = \{p \mid p \in \mathscr{P}\}$ and (2) editing distance between passwords not more than a threshold t forms an edge: $E = \{(u, v) \mid \text{lev}_{u,v}(|u|, |v|) \leq t \text{ and } u, v \in V\}$. We call the resulting graph transformed from the password dataset *password graph* G. This is a very intuitive method to turn a password dataset into graph, though there might exist some other approaches to fulfill the same goal (e.g., one can consider unique printable ASCII characters as vertices and characters forming a password are connected together, which seems reasonable but the practicability and efficiency of it remain unknown).

5.1. Construction Procedures. We describe our construction procedures of password graph with more details in this subsection. The construction procedures are organized orderly into two main phases and are illustrated in Figure 1.

Phase1. Starting with raw password dataset, we measure the frequencies of every unique password in the password dataset $\mathscr{S}$ and associate the frequency values with corresponding passwords; subsequently we remove the duplicated passwords to obtain the password dataset $\mathscr{P}$ so that two passwords picked randomly from $\mathscr{P}$ are invariably different. Every password in $\mathscr{P}$ stands for a vertex in graph G; in other words, the vertex set V for G is generated finally after this phase.

Phase2. The LD distance for every two passwords in the dataset $\mathscr{P}$ is calculated. An edge is formed if the LD value of its end vertices is not more than threshold t. We provide 3 thresholds experiment results in this paper (the selection of threshold t is an interestingly open question that needs further research). The edge set E for our graph G is generated after this phase. It is worth noting that the maximum possible number of edges is $N(N-1) \div 2$ where N is the size of set $\mathscr{P}$; it means that the resulting password graph is a *complete graph*. Another extreme case happens if there are no edges at all in G which means the resulting password graph is an *empty graph*.

In order to observe intuitively the password graph of our construction results, we have visualized the password graph with Python Graph-tool [38]. Graph-tool is an efficient Python module for manipulation and statistical analysis of graphs (aka networks). Exporting the resulting graph to Graph-tool supported file format (GraphML file format without isolated vertices). Visualizing it with some deliberate tuning of the parameters makes intricate image graph result.

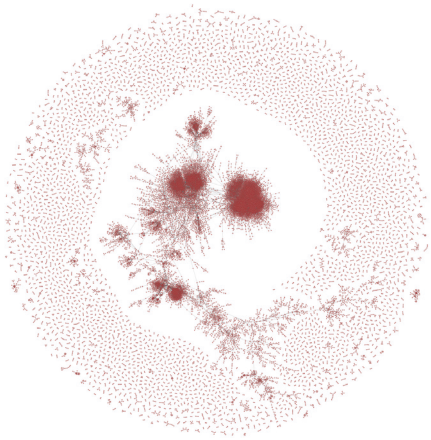


Figure 2: Visualization result of 12306.

The visualization result of 12306 is shown in Figure 2 and that of MySpace in Figure 3; the visualization results of remaining datasets are collected together in Figure 7. The internal structure and characteristic of password dataset $\mathscr{P}$ is represented modestly by its visualization image graph to some extent; some qualitative properties can be gotten from these images: a vertex in the graph G may have intensive connection with other vertices; password vertices evidently have the tendency to gather together as a cluster: 12306 has two apparently big clusters while MySpace has incalculable small clusters; vertices spreading all over the entire canvas space reflect somewhat that passwords are spreading all over the possible password space. A more in-depth quantitative analysis is performed consistently to show more inner properties of the password dataset in Section 6.

It is undeniable that the frequencies of passwords in the password dataset $\mathscr{P}$ remain untouched by us. Our final password graph is actually a simple undirected graph without any loop or parallel edge, so one feasible exploitation of password frequency may be that one repeated password attaches one *loop edge* to the vertex of the password. Besides, taking password frequency into consideration establishes more than one edge between two vertices, which is denoted as *parallel edge* in graph theory. In a word, the aforementioned operations will eventually turn our simple password graph into a *multigraph*; further augmented implementation of

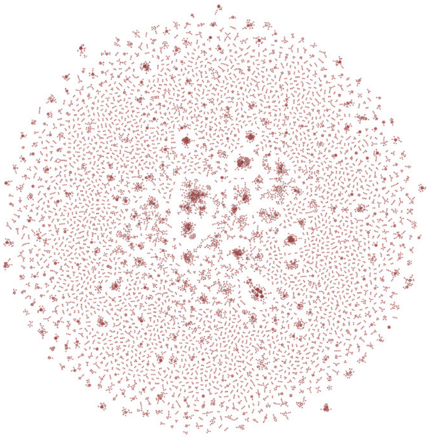


Figure 3: Visualization result of MySpace.

construction of password graph would probably embed more original and subtle properties into our final password graph but would also introduce longer time and bigger space complexities; we leave these feasible extension of implementation for further study.

6. Graph Concepts in Password Graph

Interestingly, the transformation from password dataset to password graph leads to the magical expansion of concept from graph theory into password graph. In this section, we are going to rigorously revisit the terminology in graph theory. Expanding those concepts of terminology into our refined password graph helps to analyze the password graph more concretely and quantitatively.

Degree. The degree of the vertex v, written as $d(v)$, is the number of edges with v as an end vertex. Note that there are concepts of in-degree and out-degree in directed graph, but due to the fact that our refined password graph is a simple undirected graph so there is no need to differentiate them. The concept of degree in password graph implies how many unique vertices of passwords in password dataset $\mathscr{P}$ are similar to this password; we denote degree of password p as $d(p)$.

Statistics of degree for our experimented password datasets are summarized in Table 2. The definitions of the notations used in the table are described in Notations; we also explain them in the main body of our paper. Each dataset has 3 thresholds t (from 1 to 3) experiment results exhibited in the table. One could easily notice that the column with title $\delta(G)$ is filled with zeroes; that is, the minimum degree of all password datasets is 0, meaning that there always exists isolated vertex in every password dataset; the number of passwords whose degree is zero decreases when threshold increases. Some studies have attempted to determine the number of users who appear to be choosing passwords that are random or meaningless to human beings. We think on the other side that the number of isolated vertices of passwords (no other passwords similar to it) is a more meaningful figure to achieve the same goal and is easier to evaluate. Table 3 is a segment list of passwords whose degree is zero in 12306 under threshold 3, though some of them have a common substring but they are not similar to each other actually because their number of editing is beyond the threshold and should be considered individually; frequencies of these passwords are typically 1.

Empirically, a threshold value of 3 is reasonable to analyze the experiment results, so analyses following are all based on threshold 3; one can actually get same conclusion when threshold is 1 or 2 or other viable values. Many people might hold the idea that there would definitely be more edges with bigger size of set $\mathscr{P}$. In our experiment result however, PhpBB has the most number of edges of all password graphs while Yahoo has the most number of vertices. Therefore, more vertices of password do not necessarily mean more edges. To our surprise, a password could be similar up to 3,380 passwords (in PhpBB dataset). The max degree of password increases rapidly when threshold increases; for example, with threshold 1 PhpBB's max degree is only 65, but it explodes to 3,380 when threshold is 3. More strikingly, a password is similar on average to around 150 passwords in PhpBB, even the minimum average number of similar passwords of 5 datasets is nearly 30.

One has to know that, in principle, for a specific vertex v, the possible maximum degree $\phi(t)$ of it is fixed once threshold t is determined:

$$\phi(t) = \left|S_{\text{dis1}}(t)\right| + \left|S_{\text{dis2}}(t)\right| + \left|S_{\text{dis3}}(t)\right|, \qquad (9)$$

where $S_{\text{dis1}}(t)$, $S_{\text{dis2}}(t)$, and $S_{\text{dis3}}(t)$ are the sets containing passwords whose LD distance from v is 1, 2, and 3, respectively, under threshold t. Rationally, the results of our experiment show only a lower-bound of number of similar password (the threshold we apply is smaller than the actual or typical one), the real situation may be worse than our imagination.

The distribution of password graph degree is another thought-provoking characteristic like length distribution in password datasets. Our study has shown that distribution of password graph degree correlated with its ranks sorted in descending order by its corresponding frequency is more complicated than a simple distribution. Some passwords occurring more frequently might have less number of passwords similar to them and vice versa; this is a specifically outstanding fact in 12306 that password "1qaz2wsx" has occurred 80 times but its degree is only 6; password "h123456" has 962 passwords similar to it but its frequency is only 4. Hence, the correlation between frequency of password and degree in password graph has many big fluctuations (there would be many breaks in the distribution plot). After analyzing our five password datasets of it has turned out that the degree distribution of a password graph can be approximated by the following equation:

$$\text{PGDD}(r) = k \cdot \log(r) + b, \qquad (10)$$

where PGDD(r) is the rth largest degree and the base of the log is 10.

Table 2: Statistic of experiments results.

Password dataset	$\|V\|$	t	$\|E\|$	$\Delta(G)$	$\delta(G)$	$Z(G)$	Avg. degree	$D(G)(\times 10^{-5})$	$K(G)$	$d(G)$
MySpace	41,514	1	20,119	58	0	28,119	0.97	2.33	0	$+\infty$
		2	111,884	141	0	16,748	5.39	12.98	0	$+\infty$
		3	622,087	477	0	7,290	29.97	72.19	0	$+\infty$
PhpBB	184,341	1	81,135	65	0	130,767	0.88	0.48	0	$+\infty$
		2	1,204,816	460	0	70,187	13.07	7.09	0	$+\infty$
		3	**13,836,682**	**3,380**	0	29,453	**150.12**	**81.44**	0	$+\infty$
Rootkit	56,859	1	9,281	41	0	47,093	0.33	0.57	0	$+\infty$
		2	105,275	133	0	30,564	3.70	6.51	0	$+\infty$
		3	1,062,502	953	0	15,792	37.37	65.73	0	$+\infty$
Yahoo	342,510	1	140,092	66	0	247,445	0.84	0.24	0	$+\infty$
		2	1,394,200	327	0	147,126	8.62	2.38	0	$+\infty$
		3	13,686,843	2,127	0	**77,122**	79.92	23.33	0	$+\infty$
12306	117,808	1	51,299	63	0	95,057	0.87	0.74	0	$+\infty$
		2	676,011	506	0	56,162	11.48	9.74	0	$+\infty$
		3	5,311,460	2,301	0	20,640	90.17	76.54	0	$+\infty$

Note. (1) Bold figures are the maximum value of each category for threshold 3. (2) The definitions of the notations are summarized in Notations.

Table 3: A segment of passwords with 0 degrees in 12306.

Number	Password	Frequency
1125	04768130093	1
1128	048398531	1
1138	0501154315	1
1139	0501170228	1
36909	0155402zj	1
69622	013307aini	1
120788	015210755s	1

As explained in Section 3, we employ least-squares fitting method in linear regression to approximate the distribution of password graph's degree. The fitting results of 12306 and MySpace password datasets are shown in Figures 5 and 6. The horizontal axis is the rank value and vertical axis is degree of password. The other three fitting results are shown collectedly in Figure 7. The statistics of k, b and coefficient of determination of regression results are displayed in their respective plotting figures, every linear regression (except for 12306) is with its R^2 not smaller than 0.90, which closely approaches to 1 and thus indicates a remarkably sound fitting. As for 12306 dataset, its R^2 is about 0.84, which is not an ideal one but acceptable, not as good as that of other datasets. A plausible reason may be that the 12306 dataset is collected by trying usernames and passwords from other leaked datasets online and probably cannot represent the real distribution of the entire passwords dataset of 12306. Guo et al. [34] plot the same picture as us, where the relation between log(rank) and degree is linear, which indicates the degree distribution of them follows logarithmic law but not power law. In power law, the relation between log(rank) and log(degree) should be linear.

To get the relation between password degree and crackability, we calculate each password's degree and its corresponding guess number based on [20] with sampling size 100,000. Subsequently, we plot the password's average guess number against its degree and it is shown in Figure 4: we can observe that the overall average guess number is increasing when the degree decreases or the overall average guess number is decreasing when the degree increases. Spearman coefficient for each dataset is evaluated too: the Spearman coefficient for Rootkit is −0.94, MySpace −0.88, Yahoo −0.95, 12306 −0.78, and PhpBB −0.93. The Spearman coefficient further confirms our deduction: the coefficients for all datasets are all below −0.88 except for 12306 (which might probably be caused by password data not complete); thus there is a strong relation between a password degree and its corresponding guess number: overall speaking, the bigger the password degree is, the easier it will be cracked.

Connectivity. The connectivity for a graph is the minimum number of elements that need to be removed to disconnect the remaining nodes from each other. The elements refer to vertices or edges, so there are vertex connectivity $K(G)$ and edge connectivity $\lambda(G)$. Nevertheless, neither vertex connectivity nor edge connectivity of any of our selected password datasets has the value more than 0 because of the isolated vertex in the password graph. In other words, the password graph is never connected. We only list the value of vertex connectivity in Table 2. We can further remove the isolated vertices and then evaluated the connectivity for the filtered graph; but there are still isolated clusters in the graph, so either edge or vertex connectivity is zero for filtered password graph.

Density. In simple graphs, a dense graph is a graph where the number of edges is close to the maximal number of edges. The opposite, a graph with only a few edges, is a sparse graph. For undirected simple graphs, the graph density $D(G)$ is defined

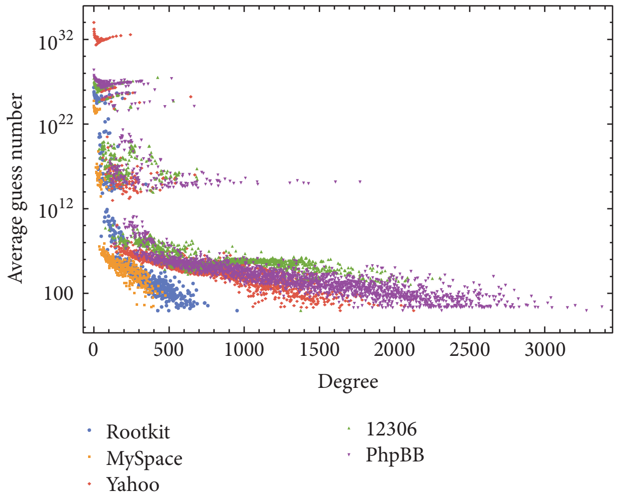


Figure 4: Average guess number against degree for all datasets.

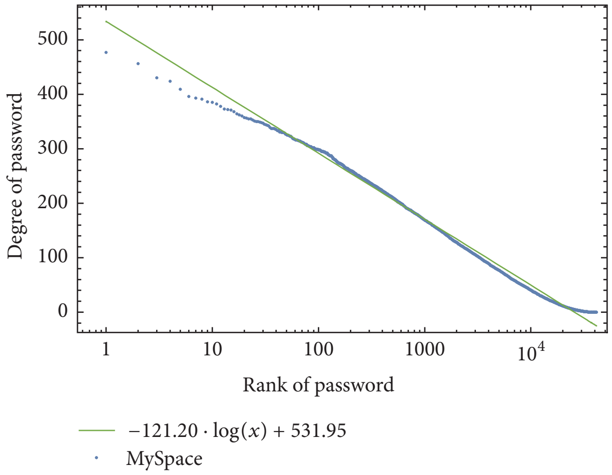


Figure 6: Fitting result of MySpace ($R^2 = 0.97$).

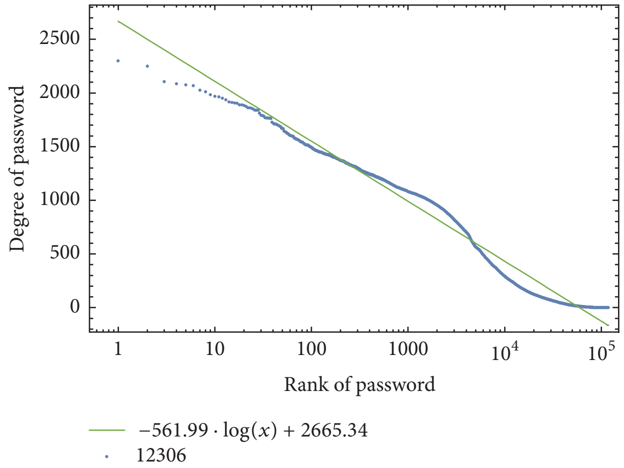


Figure 5: Fitting result of 12306 ($R^2 = 0.84$).

as follows:

$$D(G) = \frac{2|E|}{|V|(|V|-1)}, \tag{11}$$

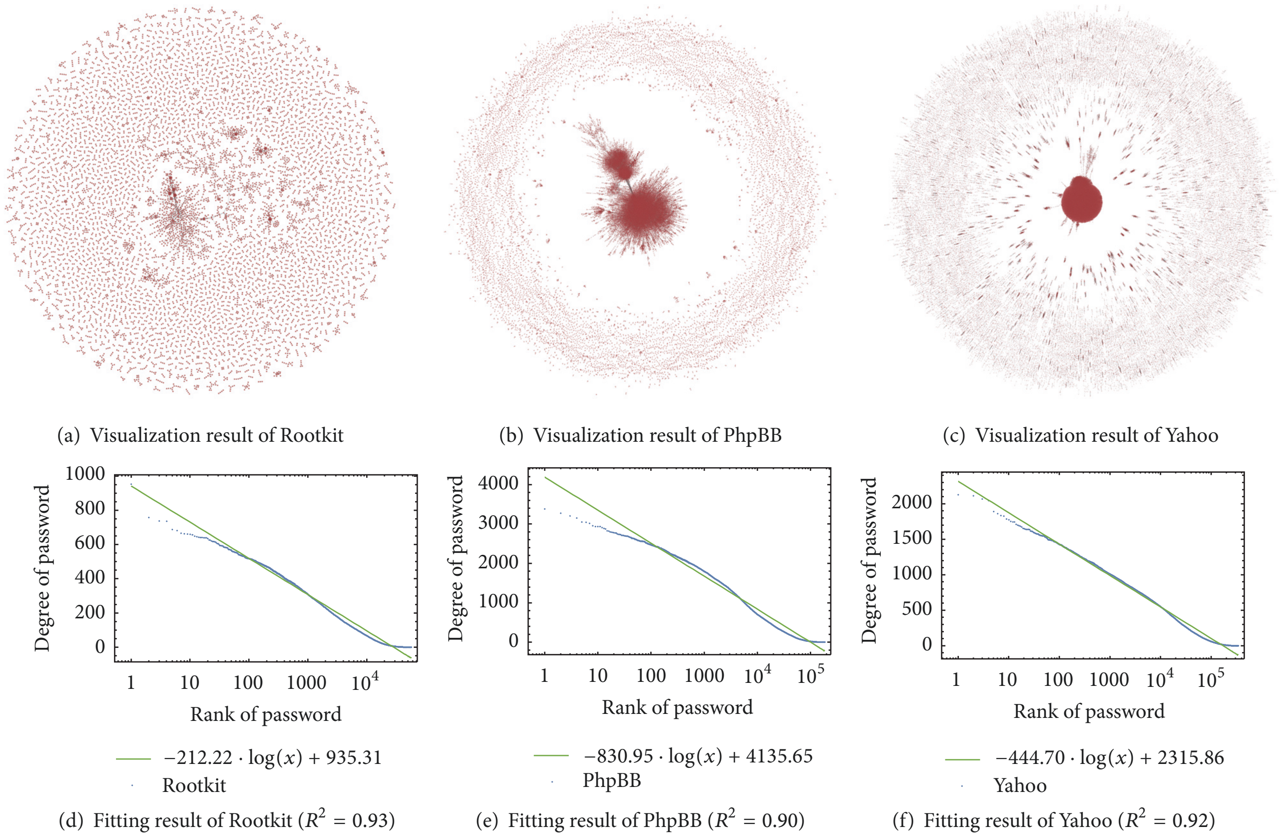
where $|E|$ and $|V|$ mean the number of edges and vertices, respectively, in the graph. The value of $D(G)$ resides between range from 0 to 1, an empty password graph has the density 0 (no edge in the graph: *empty graph*), and a complete password graph has the density 1 (any two vertices in the graph has an edge between them: *complete graph*). The density of password graph can be used as a rough approximation of password dataset's strength.

As shown in Table 2, each refined password graph is far from dense graph. We can derive the conclusion from the aforementioned *observation* that a complete password graph has the minimum password dataset strength because cracking a password successfully facilitates cracking the whole remaining passwords; an empty password graph has the maximum password dataset strength because cracking a password successfully provides no further information about the remaining passwords. Note that the exact definition of this information depends on the selection of threshold; for example, a threshold of 3 implies that a password p cracked indicates that there are passwords whose LD distance from p is smaller than 3.

Distance. The distance between two vertices u and v of a graph is the minimum length of the paths connecting them. If no such path exists, then the distance is set equal to $+\infty$.

Diameter. The diameter of a graph is the longest shortest path between any two vertices. Due to the existence of the isolated vertex, there is always at least one path of infinity length in the password graph, even if we remove the vertices with 0 degrees and calculate again; the same reason for connectivity: due to the existence of isolated clusters, the diameter for filtered password graph is still $+\infty$, and the result is also listed in Table 2. In addition to diameter, the average path length in the graph is unavoidably $+\infty$ too. Further study can be performed to evaluate the diameter and average path length for top clusters (their results will not be infinity because they are connected); we leave it for our future work.

One appealing outcome of visualization of our password graph, as we can see intuitively from Figure 2, is that vertices of password evidently have the tendency to gather together as a cluster which is a similar phenomenon in social networks. Although users choose/create passwords independently from others, they have a tendency to choose similar passwords in the mass. Apparently, there is a minority of top clusters and vast majority of nontop clusters and the top clusters' property is self-evidently of significance to us. Cluster structure in the context of networks refers to the occurrence of groups of nodes in a network that are more densely connected internally than with the rest of the network. Note the distinction between cluster and clique in graph theory: a clique is a subset of vertices of an undirected graph such that every two distinct vertices in the clique are adjacent; that is, its induced subgraph is complete but the cluster does not require the completeness property.



(a) Visualization result of Rootkit

(b) Visualization result of PhpBB

(c) Visualization result of Yahoo

(d) Fitting result of Rootkit ($R^2 = 0.93$)

(e) Fitting result of PhpBB ($R^2 = 0.90$)

(f) Fitting result of Yahoo ($R^2 = 0.92$)

FIGURE 7: Fitting and visualization results of Rootkit, PhpBB and Yahoo.

In some cases one vertex may virtually belong to more than one cluster; this might happen in a social network where each vertex represents a person, and the clusters represent the different groups of people: one cluster for family, another cluster for coworkers, one for friends in the same sports club, and so on. Nonetheless, in our password graph, we postulate that one vertex can only be part of one cluster provided that it does belong to some cluster. It is conceivable that clusters in password graph are remarkably important due to their metaproperties from the view of characteristics of clusters. Rationally, we propose the idea that a cluster in the password graph acts like a metanode in the graph; intercluster properties have more significant difference than internode properties.

Top password list [39, 40] is generally used to constitute the banned list to prevent users from choosing common passwords and circumventing the security policies. Top password structures [9] are specifically used to facilitate efficient password cracking process. We for the first time propose the concept of top clusters to outright replace top password list and top password structures for the reason that top clusters are a more accurate and precise description of weak passwords in the context of password dataset: not only passwords occurring in the top password list or its structures appear in top password structures are weak but also all the passwords belonging to the same cluster. Table 4 lists a segment of four clusters in Rootkit: clusters 1, 2 and 4 are common cluster which can be found in other password graph too; cluster 3 is only presented in Rootkit and it is a cite-specific cluster. Undoubtedly, they are all weak passwords and should be added to banned list for Internet service provider. Conclusion can be reached when combining Table 4 and Figures 2, 3, and 7 that blocking top clusters enhances the ability of the password dataset to resist trawling attack.

TABLE 4: A segment of four clusters in Rootkit.

Cluster 1	Cluster 2	Cluster 3	Cluster 4
password	123456	rootkit	qwerty
passwordx	123456x	irootkit	qwerty}
password0	1234569	1rootkit	qwertyu
Password	12345r	rootkit2	qwertyz
Password!	123456@	rootkit3	qwerty,
password0.	12345s	/rootkit	qwertyr
passw0rk	1234566	#rootkit	qwert
apassword	1234565	4rootkit	qwerty7
p4s5w0rd	1234569	rrootkit	qwerty9
p@ssword	1234561	rootKit	qwerty11
1Password!	12345678	RootKit	qwerty12

One of the multitudes of practical applications of top clusters is the building of effective mangling rules; those rules are a bit different from those used by hashcat [24] and JTR

[23]. To get the effective mangling rules from a dataset of passwords, one can compare every two passwords in the same top cluster to obtain the mangling rules. Our derived rules are actually the operations allowed in Levenshtein distance: insertion, deletion, and substitution. Substitution is the superset of Leet operation; insertion is the superset of prepending and appending. Our visualization method could also be used for password strength meter. After a user finishes entering his/her new password, we can evaluate whether it belongs to one of the existing top clusters; if it does, then we reject this password asking for a new one, else we accept it.

7. Conclusions

In this paper, we have explored possible relationships between passwords: modification-based, similarity-based, and probability-based to aid the construction of password graph. On the basis of graph theory and previous surveys, regarding passwords as vertices, we select modification-based relationship and fix a threshold value of 3 to define the edges in the graph. Our password graph enables the introduction of concepts from graph theory, including degree, density, distance, diameter, and cluster.

Password graph has some novel implications built-in for password research; we use them as an alternative analysis method for password dataset. We find surprisingly that, for the threshold 3, a password could be similar up to 3,380 passwords in the PhpBB dataset and have evidently the tendency to gather together as a cluster just like it is in a social network. Moreover, by applying linear regression we discover that password graph has logarithmic distribution for its degrees. Our findings suggest that top passwords list in adaptive password checking should be replaced with top clusters to reach a higher level of security. Ultimately, password graph could also enable the creation of effective mangling rules applied universally in dictionary attack, which only needs a large dataset of training sets without requiring a priori knowledge of user habits or personal information. Also, password graph can be used for a new kind of password strength meter.

Notations

V: Vertex set
E: Edge set
$\Delta(G)$: $\max\{d(p) \mid p \in V\}$, the maximum degree of all vertices
$\delta(G)$: $\min\{d(p) \mid p \in V\}$, the minimum degree of all vertices
$Z(G)$: $|\{d(p) \mid p \in V, d(p) = 0\}|$, the number of vertices whose degree is 0
$D(G)$: $2|E| \div (|V|(|V| - 1))$ density of the password graph
$K(G)$: Vertex connectivity
$\lambda(G)$: Edge connectivity.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Acknowledgments

The authors thank Dr. Ding Wang for his help and support. This research was supported by the National Key R&D Program of China under no. 2016YFB0800603 and no. 2017YFB1200700 and the National Natural Science Foundation of China (NSFC) under Grant no. 61472016.

References

[1] Z. Zhao, G.-J. Ahn, and H. Hu, “Picture gesture authentication: empirical analysis, automated attacks, and scheme evaluation,” *ACM Transactions on Information and System Security*, vol. 17, no. 4, article 14, 37 pages, 2015.

[2] A. K. Jain, A. Ross, and S. Pankanti, “Biometrics: a tool for information security,” *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 125–143, 2006.

[3] N. Zheng, A. Paloski, and H. Wang, “An efficient user verification system via mouse movements,” in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, pp. 139–150, ACM, October 2011.

[4] D. Florencio and C. Herley, “A large-scale study of web password habits,” in *Proceedings of the 16th International World Wide Web Conference (WWW ’07)*, pp. 657–666, ACM, May 2007.

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

[6] W. Cheswick, “Rethinking passwords,” *Communications of the ACM*, vol. 56, no. 2, pp. 40–44, 2013.

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

[8] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, “The quest to replace passwords: a framework for comparative evaluation of web authentication schemes,” in *Proceedings of the 33rd IEEE Symposium on Security and Privacy (SP ’12)*, pp. 553–567, IEEE, San Francisco, Calif, USA, May 2012.

[9] D. Wang, H. Cheng, P. Wang, and X. Huang, “Understanding passwords of Chinese users: characteristics, security and implications. CACR report,” in *Proceedings of the ChinaCrypt 2015*, 2015, http://t.cn/RG8RacH.

[10] A. Adams and M. A. Sasse, “Users are not the enemy,” *Communications of the ACM*, vol. 42, no. 12, pp. 40–46, 1999.

[11] A. Beautement, M. A. Sasse, and M. Wonham, “The compliance budget: managing security behaviour in organisations,” in *Proceedings of the Workshop on New Security Paradigms (NSPW ’08)*, pp. 47–58, ACM, Lake Tahoe, Calif, USA, September 2008.

[12] D. Wang, D. He, H. Cheng, and P. Wang, “FuzzyPSM: a new password strength meter using fuzzy probabilistic context-free grammars,” in *Proceedings of the 46th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN ’16)*, pp. 595–606, IEEE, July 2016.

[13] L. Wang, Y. Li, and K. Sun, “Amnesia: a bilateral generative password manager,” in *Proceedings of the 36th IEEE International Conference on Distributed Computing Systems (ICDCS ’16)*, pp. 313–322, IEEE, June 2016.

[14] D. McCarney, D. Barrera, J. Clark, S. Chiasson, and P. C. Van Oorschot, “Tapas: design, implementation, and usability evaluation of a password manager,” in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 89–98, ACM, December 2012.

[15] B. Ives, K. R. Walsh, and H. Schneider, “The domino effect of password reuse,” *Communications of the ACM*, vol. 47, no. 4, pp. 75–78, 2004.

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

[17] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, “Password cracking using probabilistic context-free grammars,” in *Proceedings of the 30th IEEE Symposium on Security and Privacy*, pp. 391–405, IEEE, May 2009.

[18] S. Houshmand and S. Aggarwal, “Building better passwords using probabilistic techniques,” in *Proceedings of the 28th Annual Computer Security Applications Conference*, pp. 109–118, ACM, December 2012.

[19] R. Veras, C. Collins, and J. Thorpe, “On the semantic patterns of passwords and their security impact,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS ’14)*, San Diego, Calif, USA, 2014.

[20] M. Dell’Amico and M. Filippone, “Monte Carlo strength evaluation: fast and reliable password checking,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 158–169, ACM, October 2015.

[21] R. Morris and K. Thompson, “Password security: a case history,” *Communications of the ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[22] D. V. Klein, “Foiling the cracker: a survey of, and improvements to, password security,” in *Proceedings of the 2nd USENIX Security Workshop*, pp. 5–14, 1990.

[23] John the ripper password cracker—openwall, http://www.openwall.com/john/.

[24] hashcat—advanced password recovery, https://hashcat.net/hashcat/.

[25] J. Ma, W. Yang, M. Luo, and N. Li, “A study of probabilistic password models,” in *Proceedings of the 35th IEEE Symposium on Security and Privacy (SP ’14)*, pp. 689–704, IEEE, May 2014.

[26] R. Veras, J. Thorpe, and C. Collins, “Visualizing semantics in passwords: the role of dates,” in *Proceedings of the 9th International Symposium on Visualization for Cyber Security*, pp. 88–95, ACM, October 2012.

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

[28] C. Castelluccia, M. Dürmuth, and D. Perito, “Adaptive password-strength meters from markov models,” in *Proceedings of the 19th Annual Network & Distributed System Security Symposium (NDSS ’12)*, February 2012.

[29] J. Bonneau, “The science of guessing: analyzing an anonymized corpus of 70 million passwords,” in *Proceedings of the 33rd IEEE Symposium on Security and Privacy (SP ’12)*, pp. 538–552, IEEE, San Francisco, Calif, USA, May 2012.

[30] Z. Li, W. Han, and W. Xu, “A large-scale empirical analysis of chinese web passwords,” in *Proceedings of the USENIX Security Symposium*, pp. 559–574, 2014.

[31] B. L. Riddle, M. S. Miron, and J. A. Semo, “Passwords in use in a university timesharing environment,” *Computers & Security*, vol. 8, no. 7, pp. 569–579, 1989.

[32] R. Shay, S. Komanduri, P. G. Kelley et al., “Encountering stronger password requirements: user attitudes and behaviors,” in *Proceedings of the 6th Symposium on Usable Privacy and Security*, vol. 2, ACM, July 2010.

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

[34] X. Guo, H. Chen, X. Liu, X. Xu, and Z. Chen, “The scale-free network of passwords: visualization and estimation of empirical passwords,” https://arxiv.org/abs/1511.08324.

[35] S. Perez, “Recently confirmed Myspace hack could be the largest yet,” May 2016.

[36] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics—Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[37] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, “The tangled web of password reuse,” in *Proceedings of the Network and Distributed System Security Symposium (NDSS ’14)*, vol. 14, pp. 23–26, 2014.

[38] graph-tool: efficent network analysis with python, https://graph-tool.skewed.de/.

[39] A. Vance, “If your password is 123456, just make it hackme,” *The New York Times*, vol. 20, p. A1, 2010.

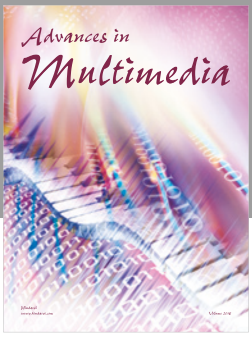
[40] S. Schechter, C. Herley, and M. Mitzenmacher, “Popularity is everything: a new approach to protecting passwords from statistical-guessing attacks,” in *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, pp. 1–8, USENIX Association, 2010.

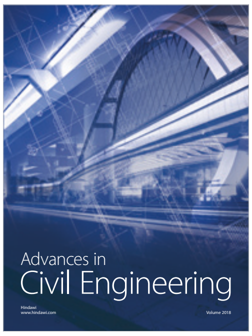
Journal of
Engineering

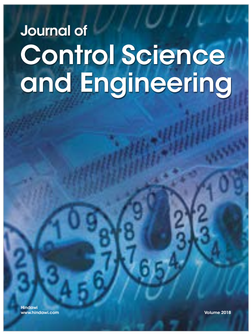
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

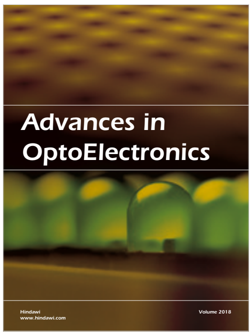

Advances in
Multimedia


Advances in
Civil Engineering


Journal of
Control Science and Engineering

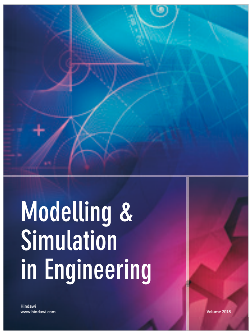
Journal of
Robotics

Journal of
Electrical and Computer Engineering

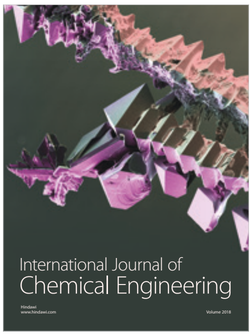

Advances in
OptoElectronics

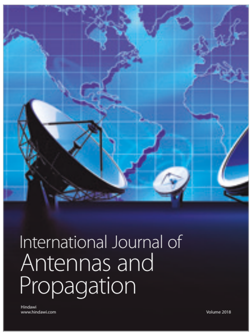
VLSI Design

International Journal of
Navigation and Observation

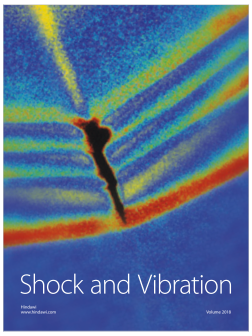

Modelling & Simulation in Engineering

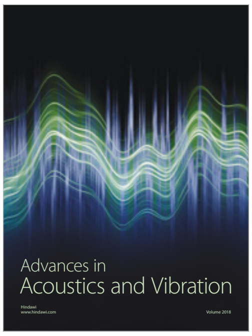
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration