# Empirical Study of Password Strength Meter Design


Yi Yang
*College of Engineering, IT and Environment*
*Charles Darwin University*
NT, Australia
vir_blade172000@163.com

*Kheng Cher Yeo
*College of Engineering, IT and Environment*
*Charles Darwin University*
NT, Australia
charles.yeo@cdu.edu.au (*corresponding*)

Sami Azam
*College of Engineering, IT and Environment*
*Charles Darwin University*
NT, Australia
sami.azam@cdu.edu.au

Asif Karim
*College of Engineering, IT and Environment*
*Charles Darwin University*
NT, Australia
asif.karim@cdu.edu.au

Ronju Ahammad
*College of Engineering, IT and Environment*
*Charles Darwin University*
NT, Australia
ronju.ahammad@outlook.com

Rakib Mahmud
*Department of Computer Science and Engineering*
*Daffodil International University (DIU)*
Dhaka, Bangladesh
rakib15-6816@diu.edu.bd



5th IEEE International Conference On Communication And Electronics Systems (ICCES 2020) -- PRE-PRINT – Will be available at IEEE Xplore



***Abstract*— Computer password was first used at the Massachusetts Institute of Technology around 1960 when researchers built a large-scale time-sharing computer called CTSS (Compatible Time Sharing System). There are many purposes where regular users require different passwords whenever they send and receive emails, do online shopping and numerous other activities on the internet. Surprisingly since the invention of the password, it has not been capable to protect the user accounts until now. There is no problem in using the similar password, but different passwords are often difficult to remember and mistakes can creep in rather easily. Many users do not know what kind of passwords should be chosen which will be strong enough to thwart all sorts of fraudulent activities. Thus, most passwords are not secure as they should be, and the users could become targets of attacks at any time. This research attempt, after a thorough literature review and in-depth empirical study, developed a software plug-in called 'Password Strength Meter', which can be used to visually inform the user about the durability of their chosen password and an estimate on the timeframe it may take to break the password using standard cracking mechanism. The output of this empirical study has been widely appreciated by the users who have tested the developed software, stating that the confidence on their chosen password increases significantly while using this tool to form a password.**



***Keywords—password, cryptography, brute force, hash, cracking***


## I. INTRODUCTION

Mainstream Internet platforms are aware of the difficulties for the users in choosing passwords and provide users with password strength meters - some of them confine the minimum number of characters in a password string; some of them use bar charts or pie charts with different colors to rank users' password strength, others place a notice describing the users about the password string composition rules. However, these password strength meters does not visually offer any indication how easily their password may be cracked, sometimes in effect make the users confused [2].

By testing a variety of different types of textual password combinations on the mainstream Internet platforms selected from Alexa Top100, it has been found that the password policies adopted by these Internet platforms are largely identical but with minor differences [2]. From the table above, it can be seen there are some discrepancies about the strength of passwords from various websites. Many studies have been conducted for many websites using the same password. Similar discrepancies are shown in which different websites display different information about the strength of the same password) This creates confusion for users and could have the illusion that their passwords are strong where in fact they are not. The most obvious finding is that most of the mainstream Internet platforms have different minimum length limits for textual password strings. The required minimum lengths of the character range from 6-8 8 bits; while some even ignore this confined limit of the character and use the algorithms to set the password strength meter differently. For instance, some use the addition algorithm system and others use the subtraction algorithm system. Apart from this some implement five colors (red, orange, yellow, green, and blue) and three colors (yellow, blue, and green) to indicate the strength ranks of users' passwords to rank the strength levels of users' passwords. Some also utilize friendly reminders to tell the users that the rules for setting strong passwords should include uppercase and lowercase letters, numbers, and symbols (at least two of them). Table 1 shows the policies adopted by different websites.

**Table 1:** Password policies of some popular websites

| Websites | Passwords | | | | |
|---|---|---|---|---|---|
| | 12345abcde | Charles123 | Char!es123 | P@ssW0rd | Re32(dqq |
| www.yahoo.com | Prompt[1] | Nil | Nil | Prompt[1] | Nil |
| www.reddit.com[2] | Red-2 | Red-2 | Green-4 | Red-1 | Orange-3 |
| www.twitch.tv | Red | Red | Red | Red | Orange |
| www.tumblr.com[2] | Red-1 | Orange-2 | Green-3 | Red-1 | Green-3 |
| www.amazon.com[3] | Nil | Nil | Nil | Nil | Nil |
| www.google.com[4] | Nil | Nil | Nil | Nil | Nil |

[1] Prompt the following: "Please create a stronger password, the one you submitted is too easy to guess"

[2] 5 levels of colour coding

3 Only displays "at least 6 characters"

4 Only display "Use 8 or more characters with a mix of letters, numbers & symbols"

**The Aim of the Research:** Better guidelines, as identified by this research, are required on ways to present the characteristics of their chosen password in a consistent manner, thus allowing users to create strong passwords that they can remember. Our research aims are mainly as following:

**1.** It is recommended that NIST (National Institute of Standards and Technology) propose and add up password

entropy as one of the password policies to the main stream Internet platforms as their password policy and use it;

**2.** To develop a JavaScript plug-in program through contemporary password-cracking algorithms and display the password durability. In addition, identify the weaker components in the given password and guide users to set a stronger password.

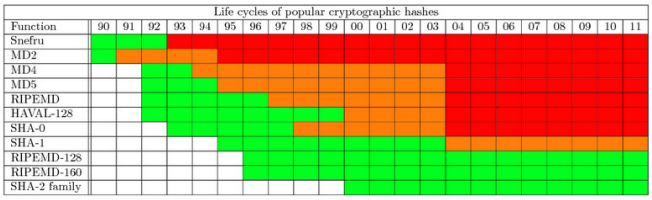
## II. Background of the Problems

This section describes some of the most common methods for password encryption.

### A. Hash Technologies

Hash encryption technology [4] is the most commonly used technique for encrypting and storing user information and passwords in computer systems and network server systems. It is a function to compress messages of any length into a fixed length message digest [6]. Also hash is a refinement of information, usually much smaller than the information, and is of a fixed length [6]. A strong hash must be irreversible, which means that any of the original information cannot be derived by the hash value. Moreover, any change in the input information will result in a significant change in the hash value, which is called the avalanche effect [6, 7]. The hash should also be anti-collision, that is, two pieces of information with the same hash value cannot be found. So these features are suitable for saving passwords meanwhile people want to use an irreversible algorithm to encrypt saved passwords [8]. Hash encryption technologies include: MD2, MD4, MD5 and SHA families. The performance of encryption algorithms can usually be measured according to the complexity of the algorithm itself, the key length (the longer the key is safer), the encryption and decryption speed, and so on [9]. Fig. 1 shows how most algorithms begins with promise, but as the year go by, eventually researchers or hackers are able to find loophole and finally crack it. The figure starts from 1990 and illustrates the situation till 2011.

**Figure 1:** Lifecycle of popular cryptographic hashes




### B. Password Cracking Algorithms

Brute Force Attack, also known as method of exhaustion, tries every combination of characters at a given length [3, 13]. This method consumes a lot of computation and is usually the least efficient way to crack hash encryption, but as long as the device runs fast enough and time permits, it will eventually find the correct password [13]. The so-called given length is depending on the password policy of target website that displays the minimum limit of the number of textual password characters when creating passwords and modifying passwords [14]. Some websites are 6-bit characters (no need to try the length less than 6 characters), most websites are 8-bit characters (no need to try the length less than 8 characters) [15].

Brute force attack tools such as 'Aircrack-ng' and 'John the Ripper' which can also constrain the type of attempts of the character. For instance, a combination of pure numbers (without trying letters and special characters), combinations of letters and numbers (without trying special characters), combinations of letters and special symbols, and so on. This can reduce many unnecessary attempts and shrink the calculation and time cost, thus reducing attack time.

Dictionary Attack is a pre-defined list of words, a collection of leaked passwords and/or an improved set of passwords as dictionary files, and guess based on certain deformation rules [16, Figure 11]. The success rate of dictionary cracking depends not only on the huge content of the word list, but also on the deformation rules [16]. For example, an attacker uses an input dictionary with a larger content, and the lesser the deformation rules applied to each word; similarly, if the attacker wants to use aggressive rules of deformation so that each word has thousands of guesses, he/she has to choose a small, more targeted input dictionary (such as P@55w0rd) [1, 4, 17]. Since most passwords are created by the users, plenty of them must contain common their personal information such as names, phone numbers, addresses and etc. [17].

### C. Password Entropy

Password Entropy is a measurement of how unpredictable a password is [19]. The password entropy is based on the character set used in the set of password combinations (include the upper case and lower case letters, numbers, and special symbols) and the length of the combination of passwords. The prediction of password entropy is the difficulty of cracking a given password through brute force attacks, dictionary attacks, and other hybrid attack methods [19]. The password entropy is usually expressed in bits, and the formula for password entropy is '$E = log_2(R^L)$', where 'E' represents the value of entropy of the password, 'R' represents the type of character (character pool), and 'L' represents the length of the password (numbers of password). If a set of passwords only contains the lowercase letter such as 'letmein', its password entropy should be $log_2(26^7)$, of which result equals to 32.9.

NIST recommends the following for the users to choose password with password entropy of 30 [3]:

**1.** At least 8 characters selected from 94 character sets, including at least one uppercase letter (char-set 26), one lowercase letter (char-set set 26), one number (char-set 10) and one special character (can be selected from the keyboard) Enter the character set 32) as shown in Table 2.

**2.** User passwords should circumvent common words or phrases as passwords, such as password blacklists.

**3.** Users should not use personal information as a password component, such as name, birthday, phone number, etc.

**Table 2:** Character-set as proposed by NIST

| Type | Character Number | Basic Elements |
|---|---|---|
| Numeric ($N$) | 10 | 0123456789 |
| Lowercase ($L$) | 26 | abcdefghijklmnopqrstuvwsyz |
| Uppercase ($U$) | 26 | ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| Other ($O$) | 32 | ~ `!@#$% ^ &*()_−+=[]\{}\|;':',./<>? |

Of course, the password entropy of a single character cannot be the only reference factor for password security. Users should also consider the length of the password and the

complexity of the combination of passwords. It turns out that too complicated short passwords are not only insecure but also difficult to remember [1]. The 8-bit complex password (MD5 encryption [4]) lasts up to 10 minutes under violent attacks [5, 18]. Some other examples have also been given in Table 3. Since password length is one of the most important factors affecting password entropy and overall strength, longer passwords can be simpler to compose, more in line with the user's memory habits, and more secure and effective [3].

**Table 3:** Example of some short password under attack

| Websites | Passwords | | | | |
|---|---|---|---|---|---|
| | 12345abcde | Charles123 | Char!es123 | P@ssW0rd | Re32(dqq |
| Password entropy | 51.7 bits | 59.5 bits | 65.5 bits | 52.4 bits | 52.4 bits |
| Brute force[1] | 4.35 days | 33 months | 192 years | 7.8 days | 7.8 days |
| Dictionary Attack[2] | instant | 0.01s | 7 hours | instant | 5 years |

[1] assuming 10 billion guesses in a second
[2] Results from online dictionary attack checker:
https://www.my1login.com/resources/password-strength-test/

Generally, password entropy gives a good indication of the strength of the passwords and correspond to the time needed to crack the password using brute force i.e. the larger the password entropy, the more time is needed to crack the password using brute force. Dictionary attacks on the other hand works differently and depends on the size of the dictionary used and may not correspond in the same manner as password entropy.

However, using both brute force and dictionary attack can give user a better indication of the actual strength of the passwords as a strong password (for a fixed length) should not be readable or just slightly modified from a readable word.)

## III. Password Policies

Though NIST in the US has proposed several guidelines, Companies sets their own policies, which is around the recommendations of NIST, but differs company to company, creating a situation of non-uniformity. Companies look for policies such as:

**1.** How long the old password can be used again;

**2.** Force the password to be changed in three months or six months;

**3.** The new password only can be changed after 3 days (restrict password changing frequency);

**4.** Allow the server administrator to track all the weak passwords and the administrator can notify the users to change their weak passwords;

**5.** Verification mechanism (mailbox verification, or SMS verification, etc.).

To sum up, those password policy differences lead to differences in the management of user passwords on different network platforms.

Password Strength Meter is an indicator, either in graphical or textual form, of the strength of a password as entered by user [2]. The mainstream network platforms give textual forms to assist the users in creating stronger passwords when the users set or modify their passwords. Some of them rank users' passwords in different colors through bar charts or pie charts which are showing as color red - weak password or invalid password, color orange - the password level is lower, color yellow - the password level is medium, color in the light blue - the password lever is stronger, color green - the password lever is safety, and color in dark blue - the password is strongest. However, as well as using bar charts and pie charts to show inaccuracies in password strength, the colors are using to show strength levels are surprisingly inconsistent.

**Table 4:** Different password meter websites

| Websites | Passwords | | | | |
|---|---|---|---|---|---|
| | 12345abcde | Charles123 | Char!es123 | P@ssW0rd | Re32(dqq |
| passwordmeter.com | 44% | 73% | 89% | 70% | 76% |
| http://password-checker.online-domain-tools.com/ | 46% | 55% | 62% | 47% | 47% |
| https://howsecureismypassword.net/ | 1 day | 8 months | 6 years | 9 hours | 9 hours |
| https://password.kaspersky.com/ | 3 mins | 3 mins | 2 months | 2 s | 12 days |
| https://www.comparitech.com/privacy-security-tools/password-strength-test/ | 4 days | 3 years | 23 years | 1 day | 1 day |
| https://random-ize.com/how-long-to-hack-pass/ | 14 days | 9 years | 609 years | 24 days | 24 days |

The table above shows multiple password meter websites that assess the strength of password. Two forms of output can be seen:

**1.** A percentage or score to rate the passwords.

**2.** The time it takes to crack the passwords using Brute Force.

Those highlighted in green are the best password among the five passwords tested and the results are consistent with all the password meter websites tested.

Those highlighted in red are the worst password among the five passwords tested and the results are not consistent with all the password meter websites tested.

Two issues can be observed:

**1.** There are discrepancies about the strength of the passwords.

**2.** A score or percentage may not be the best indication of the strength of the password. User may not correctly interpret that number and different such meters show different strength indicator for the same password which adds to the confusion.

## IV. Password Algorithm

Password Algorithm is the backend code of the different websites for reflecting the password strength meter. Basically it can be divided into three schemes [20]:

### *A. Addition Algorithm Scheme*

In scheme 1, it is an addition algorithm system which analyzes the password, combines the weight distribution, and obtains the password strength score. The higher the score, the more secure the password and the safer it is.

According to the password score, the password level is divided into seven levels from very weak (score of 0) to very safe (score >= 90). Points are given based on password length, uppercase and lowercase letters being used, number of digits and symbols used. For the shortcomings in Scheme 1, a reduction mechanism was introduced in Scheme 2. For repeated occurrences, consecutive occurrences of the character are given appropriate subtraction to make the password score more accurate. At the same time, the scoring base and calculation process of the password in the scheme 2 are very complicated. To understand the meaning of each step, please keep enough patience.

*B. Subtraction Algorithm Scheme*

The conditions to be met for the subtraction algorithm scheme is that users can only get extra points if they meet the minimum conditions.

The lowest condition is as follows:

1. The password length is no less than 8 characters;
2. Contains uppercase letters;
3. Contains lowercase letters
4. Contains numbers;
5. Contains symbols;

The minimum condition requires that item 1 be satisfied and at least any three of items 2 - 5 be satisfied.

According to the password score, the password level is divided into the following five levels from very weak (score of 0) to very strong (score >= 80). Points are subtracted if only letters, numbers or symbols are used. Points are also subtracted if there are consecutive uppercase or lower case letters.

*C. Scheme 3*

Scheme 3 which is a combo using the algorithms from scheme 1 and scheme 2, and scheme 3 is more pursuing the character string length about users setting their passwords. In another word, the scoring standard about scheme 3 is rewarding by the character string length such as more than 8 may get 2 points, more than 10 may get 3 points, more than 12 may get 5 points, more than 16 may get 10 points and 15 points are in total.

*D.* It must be acknowledged that the so-called strong password is a relative concept, and there are two major preconditions for security:

**Assumption 1.** In the worst case condition, the attacker already knows everything other than the user's original password, including all password policies, encryption algorithms, databases, etc.

**Assumption 2.** There is no absolute security on the virtual world, but if the cost of cracking the password exceeds the gains obtained, the password is relatively safe.

*Assumption 1* is often considered to be the responsibility of the users. Users often mistakenly think that their clear-text passwords are hidden somewhere on the disk. It is expected that the attacker cannot know and can only get a false sense of security. However, once the information is saved, it might be leaked, so users need to assume that the attackers had already known the information. Regardless of the information leakage of other channels, the attacker attacked the server to get all the information, or the insiders took bribes to sell the information; the user's security awareness is not enough, the set passwords are mostly composed of personal information (the combinations like name + birthday; pet name + house address; family name + phone number), the person familiar with information of himself or the attacker can easily get the user information, and the password of the user password can be obtained through certain attempts. At the same time, most users are accustomed to using the same username and password in all accounts on Internet platforms, which will cause huge losses; open virus-related emails, visit websites with Trojan viruses, computer systems without anti-virus software protection, and even computer systems security vulnerabilities, which will cause insecure user information and passwords.

*Assumption 2* is considered to be a competition between encryption technology and crack technology. Whether it is in the computer system encryption, network transmission, or sever storage encryption, *Salt encryption* is a good choice [10, 11]. Its purpose is to turn the indiscriminate attack into a targeted attack, while increasing the amount of calculation and calculation time of the attack, reducing the attack efficiency [11, 12]. The best way to add Salt is to randomly generate a string of the same length as the password. For instance, when a user A registers, the system needs to send account A(a) and password P(a) to the server. Meanwhile, the server generates a random string S(a) for the user A, and then combines them with the original password of the user A, and calculates the cipher-text P'(a) by the MD5, SHA-1, etc. discussed above (**H[S(a) + P(a)] -> P'(a)**). After saving S(a) and P'(a), the random string S(a) will remain unchanged. When another user B registers, another additional random string S(b) is generated, and the above registration process is repeated. The key here is that each user's Salt must be different [12]. Once Salt is added, even if the original password is simple, the combination is random enough. Even with the standard encryption algorithm (MD5, SHA-1), random passwords are not likely to appear in the rainbow table, and the attacker's precomputed rainbow table will be invalid. After adding salt, each user needs to perform separate calculations. At the same time, the cracking method is only a dictionary attack and a brute force attack. As long as the number of passwords is enough, the password entropy is large enough, the time required for cracking is overloaded, and such an attack becomes meaningless [10, 11, 12]. Attacks require costs which are time cost, machine cost, etc. Once the attacks costs are greater than the revenue, as a rational attacker, it is not worth doing that black hacktivism.

## V. FINDINGS AND RESULTS

A review is carried out on various websites and mapped against the responses below:

1. No explanation is given to invalid password
2. Display colour coding or equivalent
3. Prompt user with a generic guide (e.g. at least 8 characters long)
4. Gives user specific feedback to improve the password strength
5. Tells the user how long it takes to crack the password

From a user's perspective, response 1 is the worst as no feedback is given to the user about the password used. Response 2 gives the user some sense of the strength of the password but interpretation of that varies among users. Response 3 prompts the user with a generic guideline and user must meet the guidelines before the password is accepted. Response 4 tells the user exactly what is needed to improve the password strength before it can be accepted. Finally, response 5 informs the user the exact amount of time needed to crack the password. Table 5 shows the choices of responses from different websites. Security of data is as users are communicating more and more using various social networks these days [21].

**Table 4:** Different class of responses

| Websites | Reponses | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| www.yahoo.com | | | / | | |
| www.reddit.com | | / | / | | |
| www.twitch.tv | | / | / | | |
| www.tumblr.com | | / | | | |
| www.amazon.com | | | / | | |
| www.google.com | | | / | | |

| Websites (password checker) | Responses | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| passwordmeter.com[1] | NA | / | NA | / | |
| http://password-checker.online-domain-tools.com/[1] | NA | / | NA | / | |
| https://howsecureismypassword.net/ | NA | / | NA | / | / |
| https://password.kaspersky.com/ | NA | / | NA | | / |
| https://www.comparitech.com/privacy-security-tools/password-strength-test/ | NA | | NA | / | / |
| https://random-ize.com/how-long-to-hack-pass/ | NA | | NA | | / |

[1]Although a score was given by the website, it did not give the actual time needed to crack the password. This similar to colour coding but just using numbers.

From a user perspective, a consistent output from websites is important to minimise confusion. Prompts given to users with specific advice on how to improve the password strength is better than no explanation or giving the same generic response from the websites. Furthermore, informing the user the actual time needed to crack the password gives the user more useful information than colour coding.

It can be observed from the tables above there are still improvements to be made on many popular websites to enhance user experience in password creation. Password meter/checker websites on the other hand perform well from a user perspective if we disregard the discrepancies among them.

*Recommendations for password checking in websites:*

Other than the standard guidelines recommended by NIST, it should further include the following:

**1.** Gives user specific feedback to improve the password strength (response 4) – generic feedback is not entirely useful

**2.** Tells the user how long it takes to crack the password (response 5) – could be brute force and ideally combined with dictionary attack. This gives user a better sense of how strong the password is. For brute force, the computational time of a single computer is important and should be clear to the user i.e. a computer with faster computational time or if multiple computers are being used, the time taken to crack the password will be shorter. For dictionary attack, it depends on the size of the dictionary in use. The larger the size, the better it is.

## VI. Result and Discussion

This study provides an understanding that encryption technologies are becoming more sophisticated although cracking techniques are emerging endlessly. Some of the hash encryption algorithms have drawbacks as discussed before, but they are still relatively safe and widely used by increasing the length of the hash value and adding salt. In the long run, it is necessary to have a more advanced encryption algorithm (the length of the value after encryption is fixed and the algorithm is irreversible) or a new conception of storage system (like Blockchain storage [7]) to completely solve this problem. In terms of cracking methods, brute force cracking can be used to crack any password as long as it has a supercomputer or the cracking time is infinite. However, once the time required to crack a password exceeds the value of the information it protects, the brute force attack becomes meaningless. Dictionary attacks and rainbow-table attacks, although their library files are powerful enough, the hardware cost of the storage unit that needs to store this information is also multiplied. As long as the user can periodically update and update the original password, even a single character change will make the library files of the dictionary attack and the rainbow-table attack lists huge and unbearable. In addition, the mainstream Internet platforms should be more responsible for protecting the security of its users' passwords and reducing the possibility of being leaked.

After repeated passwords attempts, I found that mainstream Internet platforms have their own password policies. Although they all have NIST's password policy as the main reference, these ‘personalities’ result in a variety of password strength meters which are bar charts, pie charts, and cartoon characters dancing. Furthermore, there are also various colors that are not uniform to rate the user's password strength. These features will make the user feel ridiculous, since the password strength cannot be accurately ranked. Therefore, what is the reason why almost all mainstream Internet platforms are using similar password strength meters? It is possible that some companies' password strength meters can display the difference in user password strength relatively accurately, but such indicators do not help users to set stronger passwords either. After my numerous passwords tests, when the password contains all three types of characters and is out of order, the password character length is more than 12 bits, which is the strongest under the test of the password strength meter of any network platform. However, when it is obtained in my plug-in program, the durability is very different, which is the most fundamental purpose of my plug-in. In addition, it is promised that the plug-in will never monitor the saved user's password to expand the library file for dictionary attacks and/or rainbow table attacks. It is pleased that you would like to download the test it.

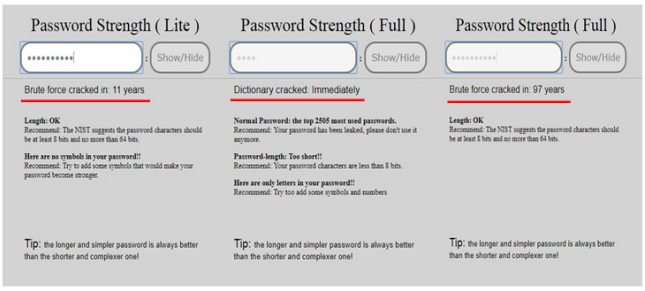
Third, in the process of making plug-in program, there are functions and algorithms that draw on some password cracking tools. ‘Aircrack-ng’ can set the length of the password strings to be cracked (2~16 digits), and password cracked types (letters, numbers and symbols), but mainly for brute force attack algorithm, does not contain dictionary attack or rainbow-table attack. Moreover, the speed of brute force cracking is not the real CPU and GPU main frequency of the current computer and/or network server. These functions have been improved in my plug-in program; ‘John the Ripper’ password-cracking tool is also mainly used for brute force attack algorithm, which includes the combine brute force attack algorithm. In addition, the attacks only calculate letters starting from a certain character of a password, some characters start to count only numbers, and some characters start with symbols. This is ridiculous, because the person using the software needs to know the approximate content of the cracked password at the beginning, such as personal name + birthday or home address + phone number, etc. For users who do not know the password approximate content, this function is very tasteless; ‘Mask attack’ mainly is a dictionary attack. When cracking a password, users can set a character to be uppercase letters, lowercase letters, numbers or symbols. Users can also constrain the length of the guessed character. This is similar function to compare with John the Ripper's combine brute force algorithm, but the actual operation effect is unsatisfactory. Because if the dictionary file of the dictionary attack does not pre-store the password combinations, the result of the crack usually fails; ‘Hashcat’ is a password attack software for the rainbow-table cracking algorithm, which requires the support of powerful library

files. The hash value of the 8 to 12-bits password encrypted by MD5 or SHA-1 without adding salt is still high by checking the rainbow-table. However, the hash value after adding salt has been encrypted through MD5, SHA-1 or other SHA family is difficult to find the password clear-text by looking up the rainbow-table. Here have other findings of this paper: 1. The hash value after adding Salt is more secure especially adding the random Salt; 2. The variable hash algorithms make the library files of the rainbow-table huge and unbearable.

In summary, the plug-in program combines two algorithms of brute force cracking and dictionary attack to make up for the shortcomings of the above-mentioned cracking tools, and is a qualitative breakthrough in clear-text password cracking.

The following are some screenshots of the functions and interfaces of this plug-in program, which visually shows the plug-in program in rating of password durability. At the same time, there are relatively objective password modification suggestions for users to set and modify passwords as well as shown in Fig. 2.

**Figure 2:** Interface for the developed Password Strength Meter, a more advanced and intuitive addition in the field




The software above has been developed based on the two recommendations above. A lite version, only showing how long it takes to crack the password using brute force and a full version containing dictionary attack in addition to brute force. Furthermore, specific feedbacks are given to the user based on what is lacking in the password.

## VII. Conclusions

To sum up, the online world has become an indispensable part of people's lives. When people access the Internet, they have to deal with clear- text passwords. Although most mainstream Internet platforms have adopted their own password strength meters to help users to set strong passwords, it turns out that these efforts are not satisfactory and counterproductive because of their own distinct password policies. When the password strength meters of users' passwords indicate inconsistently by using the exactly same password on mainstream Internet platforms, which makes users feel confused. Moreover, those password strength meters given by the mainstream Internet platforms cannot guide the users to set stronger password. This paper depends on a large number of literature survey, a variety of tests and researches on the mainstream Internet platforms. The empirical study carried out results in a body of knowledge about the weak links; the knowledge were useful in developing an advanced Password Strength Meter that boosts the users' confidence with objective insights about the chosen password. Future work involves user testing of the new password strength meter to existing ones.

## References


[1] M, Xu and W, Han, "An Explainable Password Strength Meter Addon via Textual Pattern Recognition". *Security and Comm. Networks*, 2019.

[2] C, Castelluccia, M, Dürmuth, and D, Perito, "Adaptive Password-Strength Meters from Markov Models. In NDSS". Feb. 2012.

[3] D, Pleacher (n.d.). "*Password Entropy*". [online] Pleacher.com. Available at: www.pleacher.com/mp/mlessons/algebra/entropy.html [Accessed 28 May 2019].

[4] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, M. Alazab, "A Comprehensive Survey for Intelligent Spam Email Detection," IEEE Access, 2019, 7, 168261–168295.

[5] K, Theocharoulis, I, Papaefstathiou and C, Manifavas, "*Implementing rainbow tables in high-end fpgas for super-fast password cracking*". In 2010 International Conference on Field Programmable Logic and Applications (pp. 145-150). IEEE. Aug. 2010.

[6] G, Tsudik, "Message authentication with one-way hash functions". ACM SIGCOMM *Comp. Comm. Review*, 22(5), pp.29-38, 1992.

[7] M. I. Khan, F. Faisal, S. Azam, A. Karim, B. Shanmugam, and F. D. Boer, "Using blockchain technology for file synchronization," *IOP Conference Series: Materials Science and Engineering*, vol. 561, p. 012117, Dec. 2019.

[8] B, Ross, C, Jackson, N, Miyake, D, Boneh, and J. C., Mitchell, "*Stronger Password Authentication Using Browser Extensions*". In USENIX Security Symposium (pp. 17-32), Aug. 2005.

[9] R.C., Merkle, "*A digital signature based on a conventional encryption function. In Conference on the theory and application of cryptographic techniques*" (pp. 369-378). Springer, Berlin, Heidelberg. Aug. 1987.

[10] S, Gupta, N, Goyal and K, Aggarwal, "*A review of comparative study of md5 and ssh security algorithm*". International Journal of Computer Applications, 104(14). 2014.

[11] X, Wang, L.Y., Yiqun and Y, Hongbo, "*Collision search attacks on SHA1*". 2005.

[12] M, Szydlo and Y.L., Yin. "*Collision-resistant usage of MD5 and SHA-1 via message preprocessing*". In Cryptographers' Track at the RSA Conference (pp. 99-114). Springer, Berlin, Heidelberg. Feb. 2006.

[13] D, Florêncio, C, Herley and B, Coskun, "Do strong web passwords accomplish anything?". *HotSec*, 7(6). 2007.

[14] P.G., Kelley, S, Komanduri, M.L., Mazurek, R, Shay, R, Vidas, T, Bauer, L, Christin, N and J, Lopez, "*Guess again (and again): Measuring password strength by simulating password-cracking algorithms*". IEEE Symposium on Security and Privacy (pp. 523-537). 2012.

[15] W.C., Summers and E, Bosworth, "*Password policy: the good, the bad, and the ugly*". In Proceedings of the winter international synposium on Information and communication technologies (pp. 1-6). Trinity College Dublin. Jan. 2004.

[16] E, Olson, "Robust dictionary attack of short simple substitution ciphers". *Cryptologia*, 31(4), pp.332-342, 2007.

[17] Y, Berger, A, Wool and A, Yeredor. "*Dictionary attacks using keyboard acoustic emanations*". In Proceedings of the 13th ACM conference on Computer and communications security (pp. 245-254). ACM. Oct. 2006.

[18] P, Tasevski. "*Password attacks and generation strategies*". Tartu University: Faculty of Mathematics and Computer Sciences. 2011.

[19] W, Ma, J, Campbell, D, Tran and D, Kleeman. "*Password entropy and password quality*". In 2010 Fourth International Conference on Network and System Security (pp. 583-587). *IEEE*. Sep. 2010.

[20] D, Pleacher (n.d.). "*Password Entropy*". [online] Pleacher.com. Available at: www.pleacher.com/mp/mlessons/algebra/entropy.html [Accessed 28 December 2019].

[21] Praveena, A., and S. Smys. "*Ensuring data security in cloud based social networks*". In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), vol. 2, pp. 289-295. IEEE, 2017.