Personal Information Based Dynamic Strong Password Generation

Farhana Zaman Glory*, Atif Ul Aftab† and Noman Mohammed‡
Department of Computer Science, University of Manitoba
Winnipeg, MB, R3T 2NT
Email: *gloryfz@cs.umanitoba.ca, †aftabau@cs.umanitoba.ca, ‡noman@cs.umanitoba.ca

***Abstract*—Every person using different online services is concerned with their security and privacy for protecting their individual information from the intruders. Many authentication systems are available for the protection of individuals data, and the password authentication system is one of them. But it is also necessary to ensure the strength of the password. For that reason, complex password pattern is recommended by all experts. But most of the time users forget their password because of that complex pattern. Here we come up with a unique idea of generating the passwords based on personal information. In this project, we are proposing a unique algorithm that will generate a strong password based on personal information or favorite information provided by the user. We have already designed and implemented our algorithm. We are using python programming language for the implementation. At this moment we are optimizing our algorithm and running different experiments to verify the strength of our password generator.**

***Index Terms*—Automated, hash function, generator, pattern, tokens, genetic algorithm, randomization technique**

I. Introduction

Due to the increment of information sharing, internet popularization, E-commerce transactions, and data transferring, security and authenticity have become an important and necessary subject. In this project, we have proposed to develop an automated system which will generate a strong and user-friendly password based on the given, instant and personal information by the user in the form of the combination of text and numbers. The generated password is non-guessable because of the feature of our methodology and can be used in many and different applications and internet services like social networks, secured system, distributed systems, and online services. Our proposed password generator can achieve diffusion, randomness, and confusions which are very necessary and required for the case whenever the intruder tries to crack the generated passwords. In addition to the notice that the subsequently generated passwords for the same data differ in length from one to one. The proposed work is being done by using Python programming language.

II. Motivation and Objectives

Every individual who works with different modern online services is concerned with their security and privacy for protecting their personal information from attackers. Password authentication is one of the popular authentication systems that has been used for many years for defending online accounts or services. At the same time, the user also needs to create a strong password for protecting their services from a real-world attacker. Thus it is recommended to create a unique password with a strong pattern so that they can protect it from the intruders. But usually, the users forget their password because it is not easy to remember a string with a strong pattern. So, here comes up the idea of auto-generating strong passwords based on the personal information of the user as it is easy for the user to remember the personal information which can be prompted by our proposed system from the user. Our aim is to generate a strong and non-crackable password for the online services of the user based on the personal information given by the personnel which can be remembered easily.

III. Related Work

In [1], Shay et al. found that users struggle with new and complex password requirements, and in [2], Mazurek et al. found that users who complain about complex password policies create vulnerable passwords. In [3], Huh et al. proposed a system-initiated password scheme and conducted a large-scale usability test. These works show that usability is an important factor in designing password policies. Kelsey [4] proposed a system that generates a new password by repeatedly iterating a hash function on the original master password. Abadi et al. [5] and Manber [6] proposed an approach in which password is concatenated with an arbitrary value before hashing is applied. This random value was called password supplement. Cormac et al. [7] investigated password authentication using tokens, biometrics, and authentication based on the multi-factor. Masui [8] proposed a password generation system EpisoDAS. It takes a seed string by a secret pattern drawn by the users episodic memories and creates a password. Mohammed [9] proposed a password generator using the genetic algorithm which achieves the requirements of a standard password.

IV. Proposed Work

Our aim is to develop an application that allows the user to input some personal information which is easy for them to remember. The user will be able to provide their interesting personal information that they recognized easily through our developed password generator. Inputs will vary from individual to individual and for this reason, passwords cannot be compromised easily. Input data prompted by the user will consist of different interesting information such as lucky number, favorite color, important dates etc. according to the user's choice. We will develop an efficient algorithm to generate a strong password based on the provided information so that the adversary can not penetrate the password using different cracking algorithms.

Year	Author	Background Study on Password	Password Authentication System	Password Generation	Password Crackability Test
1996	Udi Manber [6]	×	×	✓	✓
1997	Abadi et al. [5]	×	×	✓	✓
1997	Kelsey et al. [4]	×	×	✓	×
2010	Shay et al. [1]	✓	×	×	×
2012	Cormac et al. [7]	×	✓	×	×
2013	Mazurek et al. [2]	✓	×	×	×
2015	Huh et al. [3]	×	✓	×	✓
2018	Masui et al. [8]	×	×	✓	×
2018	Sura Jasim Mohammed [9]	×	×	✓	×

Fig. 1. Overview of Related Works

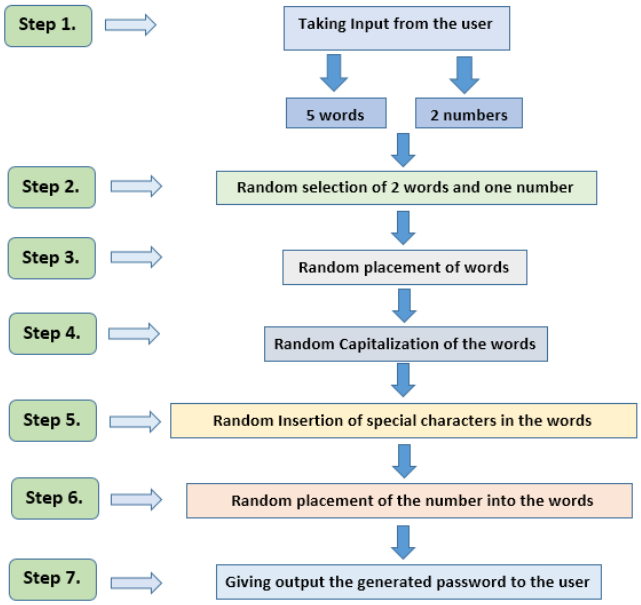

Fig. 2. Flow of the Algorithm.

V. Methodology

In our methodology, at first, we will take input information from the user who is asking for a password. The input information consists of five words and any two numbers. Both the input words and numbers are the user's most memorable or easy to remember or favorite or special ones. For the same input set the user can generate multiple passwords by our system according to their choice. If they don't like the first generated password, they can generate new passwords as many times they want. As this is a run-time system, our generator will neither save any input/outputs of the user in the memory nor maintain any log. That is why our system guarantees no risks of privacy breach.

After that, our system will randomly pick two words from five choices and one number from two choices to generate passwords. Then, the system will randomly choose the positions of the selected words. After that our password generator will capitalize any of the selected words. Then the system will append a special character to the words according to the physical similarity of the special character with the letter. For example, if a word "mango" has the letter "a" then it will be replaced by "@", as "a" and "@" are almost similar to look at but "@" is a special character including which in the password ensures more safety from the adversary and at the same time it is easier for the user to remember it as they are almost same in the outlook. Subsequently, our system will randomly insert the selected number in different permutations of the words positions either the way it is or in a reversed way. For example, if the system chooses "17" between two numbers, either "17" or "71" can be present in the generated password. In the end, there will be a random special character at the end of the generated password. These are the steps through which our algorithm work. After having input from the user our generator works on the data step by step and maintaining all the password policies this generator yields passwords for the user which can be easily remembered by the user. We are also keep tracking of the run time of the generation of passwords. For various types of inputs the run time varies but for the same input, the run time is almost the same which confirms the consistency of our system.

Input Sample 1	Password	Runtime
asix, anik, irfan, turan, oliver, 29, 12	{uran29irfan-	0.0009999275
asix, anik, irfan, turan, oliver, 29, 12	@niK29tur@n?	0.0009999275
asix, anik, irfan, turan, oliver, 29, 12	@Six29ol!ver_	0.0009999275
asix, anik, irfan, turan, oliver, 29, 12	!rfAn29an!k]	0.0009999275

Input Sample 2	Password	Runtime
mango, cat, suha, kathy, rice, 7, 81	7kathY*mango—	0.0009999354
mango, cat, suha, kathy, rice, 7, 81	18suhA-m@ngo?	0.0010002999
mango, cat, suha, kathy, rice, 7, 81	katHy@81(at]	0.0010002999
mango, cat, suha, kathy, rice, 7, 81	7cAt+r!(e—	0.0010002999

TABLE I

VI. Experimental Results

We have generated sample passwords from the same given input. A sample is shown in table 1. We also obtained run-time for generating different passwords.

VII. Tasks yet to be done

We will try to make our password generation algorithm stronger in the fitness of password policy so that our generated passwords can be hard enough to crack or guess. We will add more features and steps to our algorithm to fulfill the ultimate goal. We are using now random data sets as input data. We are still searching for real data. If we don't find out real data-sets throughout the web, we will do some survey with permission from the authority. From the survey, we can collect the real data set which we can use in our generator to generate passwords. Through using real data set we can also check how much people can remember our generated password. Through another survey, we can also get the feedback from the users whether they are able to remember the passwords generated from our system or not. We are still studying some publications on password cracking algorithm and methods. After reading them our target is to check our generator's consistency of how stronger passwords it generates. We will check by using various cracking methods and also with some of the online

password strength checkers whether our generated passwords are breakable or not. We also are planning to compare the performance of our password generator algorithm to other existing algorithms of password generation.

ACKNOWLEDGMENT

We would like to express our deep gratitude to Professor Dr. Noman Mohammed, our course instructor, for his patient guidance, enthusiastic encouragement and useful critiques of this course project.

REFERENCES

[1] Richard Shay, Saranga Komanduri, Patrick Gage Kelley, Pedro Giovanni Leon, Michelle L Mazurek, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Encountering stronger password requirements: user attitudes and behaviors. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, page 2. ACM, 2010.

[2] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 173–186. ACM, 2013.

[3] Jun Ho Huh, Seongyeol Oh, Hyoungshick Kim, Konstantin Beznosov, Apurva Mohan, and S Raj Rajagopalan. Surpass: System-initiated user-replaceable passwords. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 170–181. ACM, 2015.

[4] John Kelsey, Bruce Schneier, Chris Hall, and David Wagner. Secure applications of low-entropy keys. In *International Workshop on Information Security*, pages 121–134. Springer, 1997.

[5] Martin Abadi, T Mark Lomas, and Roger Needham. *Strenghtening Passwords*. Digital Equipment Corporation Systems Research Center [SRC], 1997.

[6] Udi Manber. A simple scheme to make passwords based on one-way functions much harder to crack. *Computers & Security*, 15(2):171–176, 1996.

[7] Cormac Herley and Paul Van Oorschot. A research agenda acknowledging the persistence of passwords. *IEEE Security & Privacy*, 10(1):28–36, 2012.

[8] Toshiyuki Masui. Episodas: Das-based password generation using episodic memories. In *Proceedings of the 2018 International Conference on Advanced Visual Interfaces*, page 73. ACM, 2018.

[9] Sura Jasim Mohammed. A new algorithm of automatic complex password generator employing genetic algorithm. *Journal of University of Babylon*, 26(2):295–302, 2018.