[ ICAETBM-2024 ]
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET

# Personalized and Memorable Password Generation to Tackle ML & AI Based Password Cracking Attacks

**Dheeraj Patil, Neeraj Tembare, Vedang Shinde, Calvin Suares, Shivam Shinde, Dr. Chandrakant Kokane**
Nutan Maharashtra Institute of Engineering and Technology, Talegaon(D), Pune

## ABSTRACT

Passwords play a crucial role in identity authentication. Password security and authenticity have become major concerns due to the increase in online information sharing, internet usage, electronic commerce transactions, and data transmission. This does, however, demonstrate that a strong password also needs to be long. Therefore, using complex password combinations is generally advised by cybersecurity experts. Users can create strong passwords with the aid of tools like password generators. But people tend to forget their passwords because of complex patterns. In this paper, we propose a novel approach to generate strong passwords, in contrast to other random password generators the passwords generated by our approach are memorable and personalized to each user instead of being a random set of words.

## I. INTRODUCTION

There are several types of authentication methods. These methods are usually divided into three main categories.

i) Knowledge-Based (What You Know): Passwords, PINs, and security questions are convenient but vulnerable to guessing or social engineering [4].

ii) Possession-Based (What You Have): Security tokens, one-time codes, and physical keys add a layer of security but can be lost or stolen[5].

iii) Biometric-Based (What You Are): Fingerprints, facial recognition, and voice recognition offer strong security but are expensive and raise privacy concerns.

Despite several such authentication methods, passwords will have and will always be the most widely used authentication method[6].

This is because password-based authentication is simple, easy to use, inexpensive, and manageable, whereas other authentication methods have a number of drawbacks, including high costs, difficulty in deploying, privacy disclosure, and so forth. Several information systems, including account login and data encryption, use password-based authentication techniques [7,8].

A majority of the internet users lack technical expertise and feel uncomfortable utilizing alternative authentication methods as which are complex and time consuming, thus they primarily rely on password-based authentication.

To safeguard their data, users generate unique passwords and security codes on their own. They frequently make simple passwords out of their names or the names of those close to them, their birth dates or other significant dates, etc. As hacking and cybersecurity breaches are becoming more common, and attackers are finding it easy to break such easily understood passwords. Although users can create unique passwords, using traditional methods[9] frequently results in weak password choices based on easily guessed information or personal details[10]. The increasing risk of AI-powered hacking exacerbates this weakness. An alarming rate of automation in machine learning algorithms allows brute-force attacks to be automated, effectively cracking weak passwords[11]. AI is also capable of predicting password patterns and developing focused attacks by analyzing user data and compromised password databases. This highlights the requirement for more advanced password-creation techniques[12]. Our proposed solution can help users stay one step ahead of these evolving AI threats by creating personalized and memorable passwords.

## II. Objective

The main goal of this paper is to analyze the existing password generation tools and algorithms. We aim to conduct a thorough evaluation of current password generation tools and algorithms, identifying their strengths and weaknesses in terms of security, memorability, and user experience.

## III. Related Work

Alphapwd: A Password Generation Strategy Based on Mnemonic Shape

The study delves into password security, proposing the Alphapwd strategy - combining mnemonic shape with password generation for creating secure and memorable passwords. The experiment results showed Alphapwd-based passwords are generally stronger against unknown attacks compared to leaked password sets.

The Alphapwd strategy, evaluated through experiments, exhibited strong resistance to unknown attacks and ease of password recall. Users could generate complex passwords that are easy to remember, enhancing overall system security. By utilizing the mnemonic shape, Alphapwd offers a practical approach to password generation, addressing the security and usability concerns typically associated with traditional password strategies.

Overall, although the Alphapwd strategy presents a promising solution to the password security dilemma, offering a unique blend of security and usability. Its innovative approach to password generation based on mnemonic shape is a bit complex and non technical users would find it time consuming and difficult to implement such techniques.

## IV. Methodology

Personalized Password Generator Algorithm

This algorithm aims to create a secure password based on a user-provided sentence or phrase, making it memorable while adhering to security best practices.

Input: The user-provided sentence or phrase (more than 8 characters)

Output: The generated secure password derived from the user's input

Working

1. Preprocessing:

- Convert the sentence to lowercase for consistency.
- Remove any spaces or special characters from the sentence. This ensures easier manipulation while maintaining memorability for the user.

2. Length Check: If the length of the sentence is less than a minimum threshold (e.g., 12 characters), perform the following:

i) Append a random selection of numbers and special characters to the end of the sentence until the desired length is reached.

ii) Ask the user to re-enter a longer prompt or ask used if he wants to “auto-fill”.

iii) If “auto-fill” is selected append a random selection of numbers and special characters to the end of the sentence until the desired length is reached[6].

3. Character Replacement:

i. Iterate through each character in the sentence with a certain probability (e.g., 30%):

ii. Replace the current character with a number (0-9)

iii. A special character from a predefined set of characters for example (!, @, #, $, %, ^ , &, *, ())

iv. The uppercase version of the current character (if it's a letter)

v. This step introduces variations and complexity into the password while keeping the base recognizable to the user.

4. Output:

Assign the modified sentence to the secure password variable.

❖ Example:

Input: "hello this is my password"

Output:"H31loT#isismYPa$$Word"

Explanation: -

a. Length Check (assuming minimum length is 12): The sentence is already 22 characters long, so no changes needed.

b. Character Replacement: Let's say the following replacements occur:

* 'h' -> 'H' (uppercase)
* 'h' -> '#' (special character)
* 'l' -> '!' (special character)
* 'o' -> '0' (number)

c. Modified Sentence

“H31lot#isismypa$$word”

d. Output:

"H31lot#isismypa$$word"

## V. Key Advantages:

Our algorithm incorporates the user's input, resulting in passwords that are more memorable by using familiar phrases or sentences, users can easily remember their passwords, eliminating the need for password managers or insecure practices like writing them down. While user-friendliness is crucial, security remains paramount. The algorithm strategically modifies the user's input through Character Replacement: Introducing variations like uppercase letters, numbers, and special characters strengthens the password against brute-force attacks. This combined approach fosters passwords that are more secure, the modifications significantly increase the difficulty of cracking passwords compared to random character strings.

Our algorithm achieves a crucial balance between user convenience and robust security. Users can create passwords that are both memorable and highly resistant to hacking attempts.

## VI. Testing and Results

It was crucial to confirm the security of our passwords, which were created by our algorithm. We adopted a multifaceted testing approach. We used popular password strength checkers on the internet, and each time our passwords were rated as "strong" or "very strong." This algorithm's superior strength comes from its deliberate use of numbers, special characters, and both uppercase and lowercase letters, going above and beyond what most checkers require. We also used crack time estimation tools, which showed that our user-generated passwords and random sets cracked much slower than expected. These programs calculate how long it will take to break a password using different methods. Our passwords' longer cracking times demonstrate the added complexity brought about by the algorithm's changes, which significantly increases their resistance to brute-force attacks and other cracking techniques. It's crucial to acknowledge that these online tools provide theoretical assessments, not representing the entire spectrum of hacking techniques. However, the results offer compelling evidence that our password generation algorithm delivers a substantial security leap compared to random password generation. This enhanced security is achieved without sacrificing memorability, as users can leverage their own personalized phrases or sentences, fostering passwords that are both user-friendly and highly resistant to hacking attempts.

It's important to acknowledge that online password strength checkers and crack time estimation tools provide theoretical assessments. While valuable, these tools don't represent the full spectrum of hacking techniques.

## VII. Conclusion

This study introduced the idea of a novel password generation algorithm that leverages user-provided phrases or sentences. The core concept lies in creating passwords that are both memorable due to personalization and secure due to algorithmic modifications. The password generator tool assists users in creating strong, unique passwords, a critical step in preventing unauthorized access to accounts and sensitive information. Unlike traditional methods that produce random strings of characters, our approach leverages user-provided phrases or sentences.

It's important to acknowledge that online password strength checkers and crack time estimation tools provide theoretical assessments. While valuable, these tools don't represent the full spectrum of hacking techniques.

Despite these limitations, the results provide strong evidence that our password generation algorithm delivers a significant improvement in password security compared to random password generation. This enhanced security is achieved without sacrificing memorability, as users can leverage their own personalized phrases or sentences.

Future Exploration

Future work can explore the relationship between user-provided customization options and the algorithm's character selection and modification strategies. This will allow us to refine the balance between password memorability and security based on user input complexity. Additionally, incorporating machine learning and artificial intelligence techniques into the algorithm holds promise for developing even stronger password generation methods. Machine learning models could be trained on large password datasets to identify patterns in compromised passwords and adjust character selection and modification strategies accordingly. AI could further enhance the process by dynamically adapting to evolving security threats and user preferences.

## VIII. References

[1] Ade-Ibijola and B. Ogbuokiri, "Syntactic Generation of Memorable Passwords," 2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)

[2] W. Jia, "Analysis on Password Attack Model and Password Generation," 2022 International Conference on Computers, Information Processing and Advanced Education (CIPAE), Ottawa, ON, Canada, 2022, pp. 145-149, doi: 10.1109/CIPAE55637.2022.00038.

[3] D. Pasquini, M. Cianfriglia, G. Ateniese and M. Bernaschi, "Reducing bias in modeling real-world password strength via deep learning and dynamic dictionaries"

[4] Kokane, C., Babar, S., & Mahalle, P. (2023, March). An adaptive algorithm for polysemous words in natural language processing. In Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems: ICACECS 2022 (pp. 163-172). Singapore: Springer Nature Singapore.

[5] Kokane, C. D., Mohadikar, G., Khapekar, S., Jadhao, B., Waykole, T., & Deotare, V. V. (2023). Machine Learning Approach for Intelligent Transport System in IOV-Based Vehicular Network Traffic for Smart Cities. International Journal of Intelligent Systems and Applications in Engineering, 11(11s), 06-16.

[6] Kokane, C., Babar, S., Mahalle, P., & Patil, S. (2022). Word sense disambiguation: A supervised semantic similarity based complex network approach. Int J Intell Syst Appl Eng, 10(1s), 90-94.

[7] Kokane, C.D., Babar, S.D., Mahalle, P.N., Patil, S.P. (2023). Word Sense Disambiguation: Adaptive Word Embedding with Adaptive-Lexical Resource. In: Chaki, N., Roy, N.D., Debnath, P., Saeed, K. (eds) Proceedings of International Conference on Data Analytics and Insights, ICDAI 2023. ICDAI 2023. Lecture Notes in Networks and Systems, vol 727. Springer, Singapore. https://doi.org/10.1007/978-981-99-3878-0_36

[8] Kokane, C. D., & Sachin, D. (2021). Babar, and Parikshit N. Mahalle." Word Sense Disambiguation for Large Documents Using Neural Network Model.". In 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE.

[9] Kokane, C. D., & Sachin, D. (2020). Babar, and Parikshit N. Mahalle." An adaptive algorithm for lexical ambiguity in word sense disambiguation.". In Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR.

[10] Kokane, C.D., Babar, S.D., Mahalle, P.N. (2021). An Adaptive Algorithm for Lexical Ambiguity in Word

Sense Disambiguation. In: Patil, V.H., Dey, N., N. Mahalle, P., Shafi Pathan, M., Kimbahune, V.V. (eds) Proceeding of First Doctoral Symposium on Natural Computing Research. Lecture Notes in Networks and Systems, vol 169. Springer, Singapore. https://doi.org/10.1007/978-981-33-4073-2_11

[11] Kokane, C., Babar, S., Mahalle, P. (2023). An Adaptive Algorithm for Polysemous Words in Natural Language Processing. In: Reddy, A.B., Nagini, S., Balas, V.E., Raju, K.S. (eds) Proceedings of Third International Conference on Advances in Computer Engineering and Communication Systems. Lecture Notes in Networks and Systems, vol 612. Springer, Singapore. https://doi.org/10.1007/978-981-19-9228-5_15

[12] D. Kokane, S. D. Babar and P. N. Mahalle, "Word Sense Disambiguation for Large Documents Using Neural Network Model," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-5, doi: 10.1109/ICCCNT51525.2021.9580101.