TokEncryption: Enhanced Hashing of Text using Tokenization

Praneeth Vadlapati
VIT-AP University, India
praneeth.18bce7147@vitap.ac.in
ORCID: 0009-0006-2592-2564

Abstract: In the current digital space, the security of sensitive information, such as passwords and private data, is of high importance. Traditional hashing methods might not adequately address data privacy concerns or vulnerabilities created due to weak passwords. Tokenization methods are utilized in natural language processing (NLP). This paper introduces a method called "TokEncryption" to utilize tokens for one-way encryption of text called hashing. A tokenizer is used to generate tokens for an input text, which are utilized to encrypt the text to create secure encrypted text. Different characters of the text are encrypted using distinct tokens to ensure a variation in encryption patterns throughout the resultant text. The process enhances data security and privacy using an unconventional approach that makes it secure from attackers attempting to reconstruct the text. The system can be used in addition to the existing encryption approaches. The results show a successful encryption of text. Weak passwords are successfully encrypted to create strong passwords with multiple types of characters, including alphabets, numbers, and special characters. The code is available at github.com/Pro-GenAI/TokEncryption.

Keywords: data privacy, data security, data encryption, natural language processing (NLP), tokenization, cryptography, information security

I. INTRODUCTION

In the digital world, the protection of sensitive data, including passwords and answers to security questions, is crucial to ensure data privacy and security [1], [2], [3]. Current encryption methods are proven to be robust in most cases [4], [5], [6]. However, the methods often fail to adequately protect the users with weak passwords or common passwords, leaving the accounts vulnerable to attacks [7], [8], [9]. Existing methods might not be sufficient to secure sensitive personal information, such as answers to security questions, which need to be stored in a more secure method. Data breaches have become common [10], [11].

A. Proposed system and its benefits

Tokenization methods are utilized in natural language processing (NLP) [12]. It is a method to convert redundant parts of text into tokens, which are numbers representing the tokens. This paper proposes a method called "TokEncryption" to leverage tokenization to enhance encryption. The method converts input text into tokens using tokenization to get tokens, which are a list of numbers generated based on the input text. The aim of this method is to make text secure before storage, to strengthen weak passwords, and to safeguard private data, ensuring data security in the event of a data breach.

B. Use cases of the system

The system is useful in cases such as password authentication, in which the password should be stored securely. Weak passwords can become stronger by utilizing this system. The system can be implemented in browsers to encrypt the password before it gets transferred to the servers. The system can supplement the encryption methods that are currently in use. The system can also be used for the validation of other values, such as security questions. The answers to the security questions could not be stored directly due to privacy concerns. The system enables the storage of private data in an encrypted method that makes it impossible to decode the text and reproduce it in its original form.

C. Related work

Encryption methods such as AES [13], RSA [14], and DES [15] are adopted worldwide [16], [17]. Existing research focuses on improving the algorithms [18], [19], [20]. However, they do not address the vulnerabilities caused by weak passwords or common passwords. Hashing and salting are commonly used to protect sensitive text, such as passwords and answers to security questions [21], [22], [23]. Multi-factor authentication (MFA) adds an additional layer of security beyond passwords [24], [25], requiring users to verify their identity through a second method. Although this improves security, it does not address the weaknesses of weak passwords or common passwords. The existing approaches leave weak passwords, common passwords, and common answers vulnerable to attacks. Existing research does not explore encrypting the text using tokens generated by a tokenizer in NLP. This paper covers the research gap by utilizing the tokenization process to generate token numbers to encrypt the text.

II. METHODS

A. Selecting and loading a tokenizer

Tokenizing a text requires a tokenizer. Tokenizers are commonly used in natural language processing (NLP) [12]. For this process, the tokenizer of the pre-trained model GPT-2 [26] is selected by considering its efficiency. The tokenizer is loaded into the system for further utilization in the steps that follow.

B. Generating tokens from text

A variety of sample text values are written for the experiment. The selected tokenizer is used to convert each text value into tokens. The tokenization process generates tokens based on redundant parts of the text, and the number of token numbers returned would be lesser than the length of the text. The sample text values used for the experiment are mentioned below.

TABLE I. SAMPLE TEXT VALUES AND THEIR LENGTH

Index	Text	Length of the text
1	Test string	11
2	password123	11
3	password	8
4	abcdefghijkl	12
5	1234567890	10
6	A longer string @ 12345	23

C. Generating secret values

Tokenized values are secured using a secret value for further enhanced security. A fixed secret number is added to the number of tokens obtained from the last step to generate a secret value for each text. The secret value can be used during encryption of each character of text in the steps that follow.

TABLE II. EXAMPLE OF A SECRET VALUE

Variable name	Variable value
Fixed number	10

Variable name	Variable value
Number of tokens	2
Final secret value	12

D. Encryption of text

The text is transformed into an encrypted format using encryption. The encryption process of a character at each index involves a method that applies a combination of the encoded character, the token at the selected index, and the secret value calculated during the last step to generate an encrypted value for the character. Each resultant character is then mapped to a readable ASCII character in the range of 33 to 126. The final characters are combined to create a new text that is readable. The procedure is repeated for all the sample text values to allow analysis using comparison with the original text. The tokens were cycled based on the position of each character to ensure that the encryption pattern varied throughout the text, which ensured there was diversity in the types of characters even if the entered text was insecure. The process ensures that each character gets transformed securely while preserving the integrity of the original text.

E. Validating the encrypted text

This step resembles authenticating a password. The entered text is encrypted using the method used in the last step. The encrypted text generated in this step is compared with the encrypted text generated in the last step. This ensures the validation of the system to resemble a use case of password authentication.

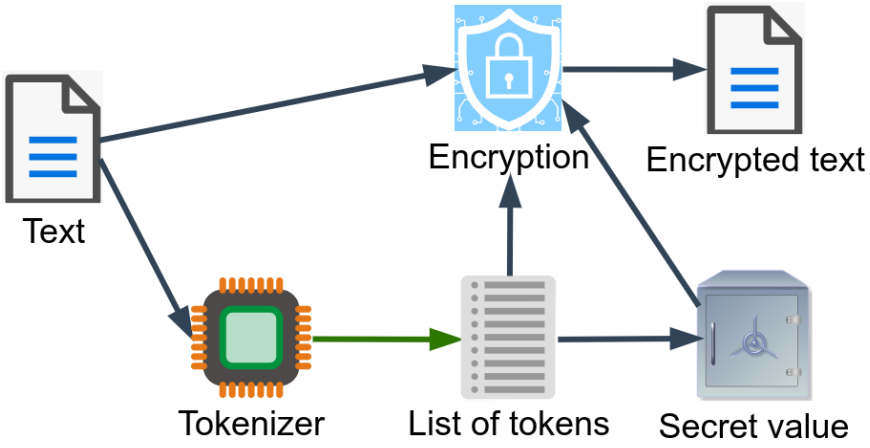

Fig. 1. Workflow of TokEncryption

III. RESULTS

A. Tokens generated from text

Tokens were successfully generated from each of the sample text values. Tokens are returned as a list of numbers. Each sample text value, tokens generated, and the number of generated tokens are mentioned below.

TABLE III. SAMPLE TEXT VALUES AND GENERATED TOKENS

Index	Text	Tokens	Number of tokens
1	Test string	[14402, 4731]	2
2	password123	[28712, 10163]	2
3	password	[28712]	1

Index	Text	Tokens	Number of tokens
4	abcdefghijkl	[39305, 4299, 456, 2926, 41582]	5
5	1234567890	[10163, 2231, 30924, 3829]	4
6	A longer string @ 12345	[32, 2392, 4731, 2488, 17031, 2231]	6

B. Generated secret values

Secret values were successfully generated for each of the sample text values based on the number of tokens generated from the text by adding it with the fixed secret number. Each sample text value and its secret value are mentioned below.

TABLE IV. GENERATED SECRET VALUES FOR EACH TEXT VALUE

Index	Text	Secret value
1	Test string	12
2	password123	12
3	password	11
4	abcdefghijkl	15
5	1234567890	14
6	A longer string @ 12345	16

C. Encrypted text

The text has been successfully encrypted by the system based on the secret values. Each sample text value and its encrypted value are mentioned below.

TABLE V. ENCRYPTED TEXT

Index	Text	Encrypted text
1	Test string	t)68@776,2*
2	password123	'K*].Y)NEyG
3	password	&t))-%(w
4	abcdefghijkl	oJ-bCtO2gHyT
5	1234567890	z`5s!d9w%^
6	A longer string @ 12345	574,N:Y,E0TE](/:}P%IXNr

D. Validating the encrypted text

Validation text has been successfully generated by encrypting each text value. The validation text generated from the text matches the encrypted text for all the test cases. This successfully resembles authentication using a password as the input.

TABLE VI. VALIDATION RESULT

Index	Text	Encrypted text	Validation text	Validation Success
1	Test string	t)68@776,2*	t)68@776,2*	Success
2	password123	'K*].Y)NEyG	'K*].Y)NEyG	Success
3	password	&t))-%(w	&t))-%(w	Success
4	abcdefghijkl	oJ-bCtO2gHyT	oJ-bCtO2gHyT	Success
5	1234567890	z`5s!d9w%^	z`5s!d9w%^	Success
6	A longer string @ 12345	574,N:Y,E0TE](/:}P%IXNr	574,N:Y,E0TE](/:}P%IXNr	Success

IV. DISCUSSION

The results demonstrate that the method can effectively encrypt text by generating diverse, secure characters from the original text. Simple text with only lowercase letters or only numbers witnessed a diversity in types of characters in the encrypted text by utilizing the system, even if only one token was generated from the original text. Diversity ensures that even predictable patterns in the original text do not lead to predictable patterns in the encrypted text, enhancing data security and privacy. However, the tokenization process consumes extra time and resources, potentially impacting the system's performance, especially when handling large volumes of requests in real-time applications that require instant authentication. Scalability may be a concern, particularly when the system is required to authenticate millions of passwords simultaneously. Improving the system's scalability and efficiency will be essential for broader adoption, particularly in high-demand environments.

V. CONCLUSION

TokEncryption presents a promising approach to the enhancement of data security and privacy by combining the strengths of tokenization and encryption. Utilizing an NLP technique called tokenization offers an unexplored way to secure and protect sensitive information. It complicates efforts by attackers by making it more complex to decode or reconstruct the original text, enhancing data safety and privacy in events of data breaches. While the current implementation of the method displayed effective results, there is a possibility for an improvement in speed, efficiency, and scalability. Future research could include making this method more secure by increasing the complexity of the process while maintaining speed and efficiency. Future research may explore increasing the length of encrypted text to ensure further security. The system could be implemented using custom tokenizers that are more scalable than the selected tokenizer. TokEncryption represents an unconventional approach to data privacy and security.

REFERENCES

[1] R. Nikam and R. Shahapurkar, "Data Privacy Preservation and Security Approaches for Sensitive Data in Big Data," Dec. 2021. doi: 10.3233/APC210221.

[2] X. Zhang, M. Xu, G. Da, and P. Zhao, "Ensuring confidentiality and availability of sensitive data over a network system under cyber threats," Reliability Engineering & System Safety, vol. 214, p. 107697, Oct. 2021, doi: 10.1016/j.ress.2021.107697.

[3] C. Wang, N. Zhang, and C. Wang, "Managing privacy in the digital economy," Fundamental Research, vol. 1, no. 5, pp. 543–551, Sep. 2021, doi: 10.1016/j.fmre.2021.08.009.

[4] M. Abdalla, M. Bellare, and G. Neven, "Robust Encryption," Journal of Cryptology, vol. 31, no. 2, pp. 307–350, Apr. 2018, doi: 10.1007/s00145-017-9258-8.

[5] P. Matta, M. Arora, and D. Sharma, "A comparative survey on data encryption Techniques: Big data perspective," Materials Today: Proceedings, vol. 46, pp. 11035–11039, Jan. 2021, doi: 10.1016/j.matpr.2021.02.153.

[6] A. Laad and K. Sawant, "A Literature Review of Various Techniques to Perform Encryption and Decryption of Data," in 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Aug. 2021, pp. 696–699. doi: 10.1109/CSNT51715.2021.9509595.

[7] R. Shi, Y. Zhou, Y. Li, and W. Han, "Understanding Offline Password-Cracking Methods: A Large-Scale Empirical Study," Security and Communication Networks, vol. 2021, no. 1, p. 5563884, Apr. 2021, doi: https://doi.org/10.1155/2021/5563884.

[8] T. V. Laptyeva, S. Flach, and K. Kladko, "The weak-password problem: Chaos, criticality, and encrypted p-CAPTCHAs," Europhysics Letters, vol. 95, no. 5, p. 50007, Aug. 2011, doi: 10.1209/0295-5075/95/50007.

[9] J. McKeon, "Weak Passwords, Poor Cyber Hygiene Invite Healthcare Data Breaches," TechTarget. Accessed: Jan. 15, 2022. [Online]. Available: https://www.techtarget.com/healthtechsecurity/news/366594983/Weak-Passwords-Poor-Cyber-Hygiene-Invite-Healthcare-Data-Breaches

[10] C. Morris, "Massive data leak exposes 700 million LinkedIn users' information," Fortune. Accessed: Jan. 15, 2022. [Online]. Available: https://fortune.com/2021/06/30/linkedin-data-theft-700-million-users-personal-information-cybersecurity/

[11] A. Holmes, "533 million Facebook users' phone numbers and personal data have been leaked online," Business Insider. Accessed: Jan. 15, 2022. [Online]. Available: https://www.businessinsider.com/stolen-data-of-533-million-facebook-users-leaked-online-2021-4?r=DE&IR=T

[12] S. J. Mielke et al., "Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP," Computing Research Repository, Dec. 2021, [Online]. Available: https://par.nsf.gov/biblio/10347731

[13] J. Nechvatal et al., "Report on the Development of the Advanced Encryption Standard (AES).," J Res Natl Inst Stand Technol, vol. 106, no. 3, pp. 511–577, Jun. 2001, doi: 10.6028/jres.106.023.

[14] X. Zhou and X. Tang, "Research and implementation of RSA algorithm for encryption and decryption," in Proceedings of 2011 6th International Forum on Strategic Technology, Sep. 2011, pp. 1118–1121. doi: 10.1109/IFOST.2011.6021216.

[15] W. E. Burr, "Data Encryption Standard," NIST. Accessed: Jan. 15, 2022. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/sp958-lide/250-253.pdf

[16] A. Hamza and B. Kumar, "A Review Paper on DES, AES, RSA Encryption Standards," in 2020 9th International Conference System Modeling and Advancement in Research Trends (SMART), Feb. 2021, pp. 333–338. doi: 10.1109/SMART50582.2020.9336800.

[17] P. Patil, P. Narayankar, Narayan D.G., and Meena S.M., "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish," Procedia Computer Science, vol. 78, pp. 617–624, Jan. 2016, doi: 10.1016/j.procs.2016.02.108.

[18] F. J. D'souza and D. Panchal, "Advanced encryption standard (AES) security enhancement using hybrid approach," in 2017 International Conference on Computing, Communication and Automation (ICCCA), Dec. 2017, pp. 647–652. doi: 10.1109/CCAA.2017.8229881.

[19] Y. Li, Q. Liu, and T. Li, "Design and implementation of an improved RSA algorithm," in 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), Jun. 2010, pp. 390–393. doi: 10.1109/EDT.2010.5496553.

[20] N. M. M. Alhag and Y. A. Mohamed, "An Enhancement of Data Encryption Standards Algorithm (DES)," in 2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Nov. 2018, pp. 1–6. doi: 10.1109/ICCCEEE.2018.8515843.

[21] S. Kharod, N. Sharma, and A. Sharma, "An improved hashing based password security scheme using salting and differential masking," in 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Dec. 2015, pp. 1–5. doi: 10.1109/ICRITO.2015.7359225.

[22] P. Gauravaram, "Security Analysis of salt||password Hashes," in 2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT), May 2013, pp. 25–30. doi: 10.1109/ACSAT.2012.49.

[23] U. Rathod, M. Sonkar, and B. R. Chandavarkar, "An Experimental Evaluation on the Dependency between One-Way Hash Functions and Salt," in 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Oct. 2020, pp. 1–7. doi: 10.1109/ICCCNT49239.2020.9225503.

[24] S. Ibrokhimov, K. L. Hui, A. Abdulhakim Al-Absi, hoon jae lee, and M. Sain, "Multi-Factor Authentication in Cyber Physical System: A State of Art Survey," in 2019 21st International Conference on Advanced Communication Technology (ICACT), May 2019, pp. 279–284. doi: 10.23919/ICACT.2019.8701960.

[25] K. Abhishek, S. Roshan, P. Kumar, and R. Ranjan, "A Comprehensive Study on Multifactor Authentication Schemes," in Advances in Computing and Information Technology, N. Meghanathan, D. Nagamalai, and N. Chaki, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 561–568. doi: 10.1007/978-3-642-31552-7_57.

[26] HF Canonical Model Maintainers, "GPT-2 [Language model]," Feb. 2019, Hugging Face. doi: 10.57967/hf/0039.