Context-based Password Cracking Dictionary Expansion Using Generative Pre-trained Transformers

Greta Imhof
School of Computing
University of Georgia
Athens, GA, United States
greta.imhof@uga.edu

Aikaterini Kanta
School of Computing
University of Portsmouth
Portsmouth, United Kingdom
aikaterini.kanta@port.ac.uk

Mark Scanlon
School of Computer Science
University College Dublin
Dublin, Ireland
mark.scanlon@ucd.ie

***Abstract*—With the rise of online criminal activity leading to the increasing importance of digital forensics, efficient and effective password-cracking tools are necessary to collect evidence in a timely manner, leading to solved crimes. Recent advances in machine learning and artificial intelligence have led to the development of context-based and large language model approaches, significantly improving the accuracy and efficiency of password cracking. This work focusses on these more modern techniques, specifically creating context-based contextual password dictionaries through training a series of PassGPTs, a large language model capable of creating password candidates from leaked password dictionary lists. This paper explores possible improvements in password cracking techniques to help law enforcement agencies in digital forensic investigations by combining PassGPT with a contextual approach.**

***Index Terms*—Password cracking, dictionary lists, artificial intelligence, large language models, context-based decryption**

I. Introduction

Password security is essential to keep personal information and accounts private. However, with the increased focus on security, digital forensics is becoming a more vital part of law enforcement investigations. Online criminal activity is now more common than ever, and password-protected encrypted data are often a key barrier to evidence collection. Traditional password cracking techniques, such as brute force and dictionary attacks, have been used by law enforcement agencies for a while, but are often inefficient and time-consuming when time is an important factor in investigations. Law enforcement agencies need stronger, more efficient tools to keep up with the growing rate of cybersecurity threats and the improvements to digital security. Obtaining access to encrypted data is one of the most commonly needed techniques by law enforcement during lawful investigation under warrant [1]. Currently, advances in machine learning and artificial intelligence approaches are leading in the field of password cracking. This research explores both traditional and modern approaches to password cracking, specifically diving into context-based password cracking, as well as using large language models to aid the guessing process.

A. Contribution of this Work

The work described in this paper shows the effectiveness of combining a strategic contextual approach with the power of large language models in the password cracking process. This work, which builds on previous work by [2, 3, 4], aims to assist law enforcement agencies in criminal investigations by providing a password cracking method that allows extensive lists of password candidates that are specific to the traits or interests of a subject to be generated quickly. The effectiveness of this approach is evaluated on the basis of its performance on ten datasets of leaked passwords from websites associated with a particular topic. The evaluation shows that this method is capable of producing correct password candidates that are not generated by other dictionary datasets. The contribution of this work includes the following:

- A detailed description of the methodology used for this approach to password cracking, including the dataset selection process along with the process of training models and generating contextual password dictionaries.
- An in-depth results section, which analyses the effectiveness of the approach on ten datasets based on various factors.
- A comprehensive discussion on the impacts and uses of this approach to password cracking, as well as strategies to further improve and expand upon this method in future research.

The remainder of the paper is organised as follows: Section II presents a brief review of the literature on related work in password cracking and large language model approaches. Section III details the methodology of the approach, including dataset selection, PassGPT training, dictionary generation, and the application of mangling rules. Section IV presents the results of the tests, comparing the success rates of password cracking and the strength of cracked passwords between custom PassGPT-generated dictionaries and a general baseline dataset. Finally, a discussion of the results and a conclusion and possible avenues for future work are outlined in Sections V and VI respectively.

II. Background and Related Work

A. Password Cracking Techniques

Over the years, there have been many approaches to password cracking, each with their unique advantages and disadvantages. The most direct approach to password cracking is a brute-force attack, or exhaustive search. This method involves testing every possible combination of letters, numbers, and special characters within a certain maximum password length. Although guaranteed to work, the downside of this method is the variable of time. Shorter passwords can be fully tested in a reasonable amount of time, but longer passwords are not guaranteed to crack in a suitable time frame for a forensic investigation [5].

A second commonly used approach to password cracking is a dictionary attack. Using this technique involves testing each entry in a given wordlist. An example of a wordlist that is often used is the RockYou dataset, which consists of about 14 million unique passwords, coming from more than 32 million accounts and acquired from a data leak in 2009 [6]. In addition to testing the exact versions of the passwords from the dataset, password mangling rules, such as adding numbers or substituting special characters, capitalising the first letter, or replacing an 's' with an '$' are applied to create different variations of the passwords. This is done to mimic how people create their own passwords, especially if asked/urged to create more "complex" passwords [7]. Although this approach is less time intensive than a brute-force attack and is one of the go-to approaches nowadays, it is not guaranteed to work, as word lists do not contain every password possibility.

Rainbow tables are another approach to password cracking that focusses on efficiency. Rainbow tables involve storing precomputed hashes of passwords and using reduction functions to transform the hashes back into passwords to test them [8]. The tables do not require a large amount of storage because only the beginnings and ends of the password chains are stored. However, a downfall of this method is shown when salts, random values added to a password before hashing it, are used, as their presence would require a separate rainbow table for each possible salt, which makes the use of salting a sufficient mitigation technique against rainbow tables [9].

Most modern approaches to password cracking are based on machine learning approaches. These techniques mostly involve producing a list of candidate passwords based on a given input and using mangling rules to give variation to these results. Machine learning models are beneficial in the efficiency and effectiveness of password cracking due to their ability to recognise patterns as well as their ability to be automated. This can be exemplified by neural networks like PassGAN, which uses a Generative Adversarial Network to create password candidates that mimic human passwords based on the distribution of real passwords from data leaks. [10]. The following sections of this paper will focus on the two approaches of using context-based dictionaries and the use of large-language models to produce more accurate and efficient results of password attacks.

B. Context-Based Dictionary Attacks

While passwords have become more secure over the years, in part due to websites with length and complexity requirements to boost security, user-chosen passwords are far from random. Since the number of passwords the average person has is constantly increasing, users tend to use weak passwords that are easy to remember or reuse the same stronger passwords over and over again with slight modifications [11, 12]. Although this leads to less secure accounts, it is beneficial in the case of digital forensics, as if one variation of the password is cracked, it is much easier to figure out subsequent passwords. According to a study that analysed 70 million passwords, dictionaries specific to certain demographics, such as age, language, and profession, in general outperformed generic dictionaries in their success rates [13]. This shows how demographic information on a subject has an effect on the passwords that they choose, and therefore would be beneficial to utilise in password-cracking operations.

Context-based password-cracking techniques involve using contextual information about a target to create personalised password dictionaries. Using information on a subject's job, interests, hobbies, etc., there is an increased opportunity to crack a user's password or passwords. In a study that used a dataset of leaked passwords from the Qatari National Bank, AlSabah et al. [14] found that 3.9% of users included their phone number, names, or personally relevant dates in their password. A similar study focussing on the differences between Chinese and English passwords found that the number of Chinese users who included their phone number in their password was 2.9% [15]. The benefits of using a semantic approach to password cracking are further highlighted by Kanta et al. [2] as customised dictionaries around specific topics can recover passwords that generic dictionary attacks cannot. In fact, in this and subsequent studies, the authors showed that while larger and more generic dictionary lists such as RockYou and Ignis-10M[1] managed to find more passwords in general, for passwords that were deemed difficult to crack by the password strength estimator *zxcvbn*[16], the improvement of adding a contextual dictionary to these generic ones resulted in an improvement of up to 50% [4, 17]. This shows that a context-based approach is beneficial for password cracking, especially if used in conjunction with a more general dictionary attack approach.

C. Large Language Model Approaches

A language model is a machine learning model that deals with performing language-related tasks. Its main purpose is to predict word sequences or generate new text when prompted with any given input. Large language models are more advanced versions of standard language models. With more parameters, they are significantly better performing in their capability for text generation and have a wider variety of possible applications. Some important aspects of large language models are in-context learning, task demonstration-based model

[1] https://github.com/ignis-sec/Pwdb-Public

training, and human feedback-based reinforcement learning, using human feedback as a way to train and improve model performance [18]. Some current uses of large language models include translations, summarisations, answering questions, and code generation.

Large language models have been exceptionally useful in the field of cybersecurity. Some of their uses include anomaly detection to discover potential threats, analysis of large amounts of textual data, such as security logs, and education against cyberattacks[19]. Although large language models can be useful tools in terms of cybersecurity defence, it is also important to recognise the potential for misuse. There exist models designed specifically to cause harm to others. An example of this is WormGPT, which is a tool that generates personalised phishing emails to be used maliciously [20].

A specific approach to password cracking using a large language model is a model called PassGPT [21]. PassGPT is a password-guessing and password-strength estimation tool created using the GPT-2 architecture. It was trained using a large amount of leaked passwords coming from leaked password datasets, e.g., RockYou. PassGPT was trained using two training sets, one with unique passwords only and one with all occurrences of passwords, and performed better when trained in the unique password training set [21].

PassGPT generates passwords using an approach called guided password generation. This approach works by modelling each token, or character, individually, as opposed to older models that generate passwords as a whole. This approach allows users of the model to specify constraints such as length, character placement, or password structures. This is beneficial when the requirements used to create the targeted passwords are known [21]. Although a powerful tool, there are still potential faults in this process. Word truncation is a common occurrence when using PassGPT, as the model considers the given pattern requirement, such as five letters followed by three numbers, over the model's prediction of the next character [22].

III. Methodology

In order to determine the benefit of combining the capabilities of PassGPT with the strategic approach of contextual dictionary generation, the process involved training PassGPT models with contextual dictionaries and subsequently using these models to generate password candidates. The generation of context-based dictionaries involves the use of seed words and a structured hierarchical version of Wikipedia, DBPedia, to generate dictionaries of related words, a process described in detail by Kanta et al. [4]. With these generated candidates, Hashcat[2], an advanced password recovery tool, was used to mangle them and test the accuracy of the guesses.

A. Baseline Selection

As a baseline to compare the results of the custom PassGPT-generated dictionaries, the Ignis-10M dataset, consisting of 10 million real passwords, was used. Created in 2020, this password dataset comes from various data leaks prior to that year. It was chosen because it is relatively recent compared to other password-cracking dictionaries, the passwords are in plaintext form rather than hashed form, and it is a general list as opposed to the custom PassGPT model-generated lists that are topic-specific. Because of this, comparing the newly generated dictionaries with the Ignis-10M dataset shows the effects of a context-based approach combined with PassGPT.

B. Dataset Selection

Ten datasets were selected to match those by Kanta et al. [17] to test the proposed password cracking method. Each dataset, originating from hashes.org, represents a different hobby or topic of interest such as cars, music, cooking, or video games. Using leaks from websites related to these topics, each dataset contains passwords from real users, ranging in size from 25,271 to 42,908,386 passwords in each dataset. The contents of each dataset are represented in plaintext and do not contain any repetition. These datasets were chosen because they each range in subject focus as well as size, so they provide a wide variance of the data.

C. Password Cracking Methodology

The first step in the proposed approach is to gather context-based dictionaries to train PassGPT. The second step is to train the PassGPT models using the contextual dictionaries as training data. This creates a custom context-based PassGPT model that can generate password candidates of similar likeness to the training data, tailored to a specific topic. For each of these ten password dictionaries, a custom PassGPT model was produced.

Model training took between approx. 40 minutes to more than 32 hours to complete on a typical desktop machine. The amount of time it took to train each model was directly correlated with the size of the training data, a factor to take into account in the case of a timely digital forensic investigation. From this, questions arise on the payoff between larger training datasets compared to faster model creation times.

With the trained PassGPT model, the next step is to generate a new password dictionary to be used for password cracking. With the PassGPT model, it is possible to generate any number of candidates, allowing a quick way to generate large amounts of potential passwords. After the training of the custom PassGPT models was completed, 10 million passwords were generated for each of the datasets to stay consistent with the Ignis-10M dataset for comparison purposes. This process required only 30 minutes to an hour to generate an extensive dictionary of potential password candidates.

After generating the desired number of password candidates, mangling rules were used to diversify the passwords and increase the likelihood of getting a hit on a password. These mangling rules change the password candidates in a multitude of ways, including adding numbers and special characters, capitalising letters, and swapping similar characters, such as 's' for a '$'. After custom lists of 10 million passwords were

[2]https://hashcat.net/hashcat/

Dataset	Dataset Size	Seed Word	Custom Training Dictionary Size
Battlefield	419,940	Battlefield	415,311
JeepForum	239,347	Car	853,825
EverydayRecipes	25,271	Cooking	524,269
Wattpad	23,531,304	Fanfiction	641,007
MangaTraders	618,237	Manga	180,641
Minecraft	143,248	Minecraft	243,803
AxeMusic	252,752	Music	1,001,173
Wanelo	2,130,060	Shopping	627,487
DoSportsEasy	46,113	Sports	31,918
Zynga	42,908,386	Zynga	443,443

TABLE I: The sizes of the leaked password datasets and the sizes of their corresponding custom dictionaries.

Dataset	Ignis-10M	Custom PassGPT	Custom PassGPT Excl.	Custom PassGPT Impr.
Battlefield	57.92%	11.77%	0.36%	0.62%
Car	45.05%	14.32%	0.34%	0.75%
Cooking	61.20%	22.05%	0.49%	0.80%
Fanfiction	15.58%	2.15%	0.10%	0.64%
Manga	52.21%	10.77%	0.55%	1.05%
Minecraft	35.39%	3.56%	0.14%	0.68%
Music	40.88%	9.58%	0.36%	0.88%
Shopping	39.32%	6.25%	0.35%	0.89%
Sports	40.94%	5.23%	0.11%	0.27%
Zynga	16.10%	1.98%	0.18%	1.12%

TABLE II: Total passwords cracked as a percentage of total passwords in the dataset. The Custom PassGPT Excl. column contains passwords found uniquely by the PassGPT model and the Custom PassGPT Impr. column shows the improvement over Ignis-10M provided by the PassGPT dictionaries.

generated, hashcat was used to apply the mangling rules to each of the dictionaries and to the Ignis-10M dataset. The ruleset of mangling rules that was used was the best64 ruleset, which is a collection of 64 common mangling techniques that a person would use in order to either make their password feel more secure or to fulfil password requirements set by websites when creating passwords for new accounts. This ruleset was selected for its time efficiency while also providing enough variation in the password candidates to be effective. From the original 10 million passwords in each dataset, the best64 rule created dictionaries of 770 million passwords for each of the original dictionaries, including some duplicates that were subsequently filtered out.

The final step in this process is to use the final contextual PassGPT generated and mangled dictionaries to attempt to crack passwords. This can be done either on individual passwords or on a dataset of passwords, such as in the case of this research. For this step, the ten datasets of leaked passwords from different websites were used to evaluate the performance of their corresponding custom PassGPT-generated dictionaries. The ten leaked password datasets were also used to evaluate the performance of the mangled Ignis-10M dataset. The results of these evaluations can be seen in Table II.

IV. Results

After completing the process of cracking passwords using custom PassGPT-generated dictionaries, the results were analysed in two ways: by the sheer number of passwords cracked and by the strength of passwords that were cracked. The baseline for comparing the results of custom PassGPT-generated dictionaries is the result of the Ignis-10M dataset.

A. Number of Cracked Passwords

Table II shows the percentage of total passwords cracked by Ignis-10M as well as the percentage of total passwords cracked by custom PassGPT-generated dictionaries for each of the datasets. On average, Ignis-10M cracked about 40.45% of all passwords in the datasets. In contrast to this, custom PassGPT-generated dictionaries cracked an average of 8.76% of all passwords in the datasets. Although Ignis-10M was able to crack a greater number of passwords, the Custom PassGPT exclusive column gives a reference of how this method of generating context-based password-cracking dictionaries is beneficial. This column shows the percentages of passwords found by the custom PassGPT dictionaries that were not found by Ignis-10M. This resulted in an average of 0.30% of the total passwords being found exclusively by the custom PassGPT dictionaries and not the Ignis-10M dataset. An average improvement of about 0.70% results in a significant amount of more passwords being cracked, as the datasets contain hundreds of thousands to tens of millions of leaked passwords.

B. Strength of Cracked Passwords

The second way to analyse the results is based on how strong the passwords were that were cracked by each of the dictionaries. To do this, `zxcvbn`[3], a password strength

[3]https://github.com/dropbox/zxcvbn

Dataset	Ignis-10M	Custom PassGPT	Custom PassGPT Excl.	Custom PassGPT Impr.
Battlefield	19,514	299	58	0.30%
Car	4,090	73	15	0.37%
Cooking	327	8	2	0.61%
Fanfiction	355,699	10,041	1,550	0.44%
Manga	27,375	563	127	0.46%
Minecraft	3,950	59	13	0.33%
Music	5,905	208	52	0.88%
Shopping	59,766	1,489	334	0.51%
Sports	1,094	7	2	0.18%
Zynga	574,279	14,257	2,732	0.48%

TABLE III: Class 3 passwords. The Custom PassGPT Excl. column contains passwords found only by the PassGPT model and the Custom PassGPT Impr. column shows the improvement over Ignis-10M provided by the PassGPT dictionaries.

estimator, was used. `zxcvbn` classifies passwords into five classes, with Class 0 passwords being the weakest and Class 4 passwords the strongest. There were not enough Class 4 passwords found to make any conclusions about the custom PassGPT generated dictionaries' effects, however, the number of Class 3 passwords, the second-strongest class of passwords, showed improvement. Table III shows the results of the number of Class 3 passwords cracked by Ignis-10M, together with the number of Class 3 passwords cracked by custom PassGPT-generated dictionaries for each of the datasets. The Custom PassGPT Exclusive column shows the number of Class 3 passwords that were found by custom PassGPT-generated dictionaries and were not found by Ignis-10M. The data show an average of 0.45% improvement in the number of Class 3 passwords found by the custom PassGPT-generated dictionaries. This is important in the case of forensic investigations, as relevant evidence is more likely to be protected by passwords in Classes 3 and 4 as opposed to those in lower and weaker classes.

V. Discussion

The results of this research study show that the use of custom PassGPT models is capable of producing successful password candidates that the Ignis-10M dataset does not contain. Although custom PassGPT models do not exceed a large dataset of real passwords, such as Ignis-10M, in terms of the total number of passwords cracked, they show improvement when used in conjunction with it. These findings have important implications in terms of forensic investigations, suggesting that incorporating a contextual approach with the capabilities of a large language model can lead to an improved success rate in password cracking attempts. However, there are limitations to this research, including the limited number of datasets used. Although some datasets showed promising results, others were not as successful. More research should be done using a wider variety and a greater number of password databases to support these findings.

VI. Conclusion

The potential use for smarter context-based password cracking dictionaries in digital forensics is great. In many law enforcement investigations, whether the crime was committed online or not, online information about a subject can provide valuable information and evidence to a case. Because much of this information is protected in accounts with passwords, it is essential to get around them promptly. Currently, many law enforcement agencies do not have the funding, resources, and strategies necessary to crack passwords in a timely and efficient manner. This is why this research aims to provide an improved method of cracking more passwords than dictionary lists alone. For this research, PassGPT models were trained with contextualised dictionaries to generate topic-specific password candidates. A comparison of passwords cracked by custom PassGPT models with the Ignis-10M dataset showed an improvement in both the number of cracked passwords and the strength of the cracked passwords. This shows how combining the strengths of large language models with a contextual approach allows for higher chances of cracking passwords when used in a forensic investigation.

A. Future Work

As mentioned previously, the time necessary to complete the custom PassGPT model training process, ranging from under one hour to over 32 hours, raises questions about the payoff between larger training datasets and faster model creation times. When considering the efforts of law enforcement agencies in forensic investigations, both time and accuracy are important factors. As part of future work related to this research, the effects of the size of the training data on the accuracy of the model's predictions will be taken into account.

Another aspect of this method of password cracking that should be taken into account is the mangling ruleset used on the generated password dictionaries. There are other mangling rulesets that exist that are much more extensive than the best64 ruleset. Using a different mangling ruleset on the PassGPT-generated password dictionaries could potentially result in a higher number of passwords cracked; however, a longer ruleset would increase the total time required to complete the password-cracking process.

A final factor to consider in future work is how the context-based custom PassGPT models would compare to a general custom PassGPT model, as well as the default PassGPT model. For example, training a custom PassGPT model on the Ignis-10M dataset and using that as a comparison to the results

of the context-based custom PassGPT models and the default PassGPT model would further show the impact of a contextual approach to password cracking.

References

[1] C. Hargreaves, F. Breitinger, L. Dowthwaite, H. Webb, and M. Scanlon, "DFPulse: The 2024 Digital Forensic Practitioner Survey," *Available at SSRN 4954821*, 2024.

[2] A. Kanta, I. Coisel, and M. Scanlon, "A novel dictionary generation methodology for contextual-based password cracking," *IEEE Access*, vol. 10, pp. 59 178–59 188, 2022.

[3] ——, "PCWQ: A Framework for Evaluating Password Cracking Wordlist Quality," in *Digital Forensics and Cyber Crime*, P. Gladyshev, S. Goel, J. James, G. Markowsky, and D. Johnson, Eds. Cham: Springer International Publishing, 2022, pp. 159–175.

[4] ——, "Harder, better, faster, stronger: Optimising the performance of context-based password cracking dictionaries," *Forensic Science International: Digital Investigation*, vol. 44, p. 301507, 2023.

[5] M. Raza, M. Iqbal, M. Sharif, and W. Haider, "A survey of password attacks and comparative analysis on methods for secure authentication," *World Applied Sciences Journal*, vol. 19, no. 4, pp. 439–444, 2012.

[6] M. Zhang, Q. Zhang, X. Hu, and W. Liu, "A password cracking method based on structure partition and bilstm recurrent neural network," in *Proceedings of the 8th International Conference on Communication and Network Security*, 2018, pp. 79–83.

[7] S. Aggarwal, S. Houshmand, and M. Weir, "New technologies in password cracking techniques," *Cyber Security: Power and Technology*, pp. 179–198, 2018.

[8] F. Yu and Y. Huang, "An overview of study of passowrd cracking," in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*. IEEE, 2015, pp. 25–29.

[9] P. Oechslin, "Making a faster cryptanalytic time-memory trade-off," in *Advances in Cryptology - CRYPTO 2003*, D. Boneh, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 617–630.

[10] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, "Passgan: A deep learning approach for password guessing," in *Applied Cryptography and Network Security: 17th International Conference, ACNS 2019, Bogota, Colombia, June 5–7, 2019, Proceedings 17*. Springer, 2019, pp. 217–237.

[11] A. Kanta, I. Coisel, and M. Scanlon, "A survey exploring open source intelligence for smarter password cracking," *Forensic Science International: Digital Investigation*, vol. 35, p. 301075, 2020.

[12] A. Das, J. Bonneau, M. Caesar, N. Borisov, and X. Wang, "The tangled web of password reuse." in *The Network and Distributed System Security (NDSS) Symposium*, vol. 14, no. 2014, 2014, pp. 23–26.

[13] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *2012 IEEE Symposium on Security and Privacy*. IEEE, 2012, pp. 538–552.

[14] M. AlSabah, G. Oligeri, and R. Riley, "Your culture is in your password: An analysis of a demographically-diverse password dataset," *Computers & Security*, vol. 77, pp. 427–441, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818302979

[15] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: Understanding passwords of chinese web users," in *28th USENIX Security Symposium (USENIX Security 19)*. Santa Clara, CA: USENIX Association, Aug. 2019, pp. 1537–1555. [Online]. Available: https://www.usenix.org/conference/usenixsecurity19/presentation/wang-ding

[16] D. L. Wheeler, "zxcvbn: Low-Budget password strength estimation," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, Aug. 2016, pp. 157–173. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/wheeler

[17] A. Kanta, I. Coisel, and M. Scanlon, "A comprehensive evaluation on the benefits of context based password cracking for digital forensics," *Journal of Information Security and Applications*, vol. 84, p. 103809, 2024.

[18] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 15, no. 3, pp. 1–45, 2024.

[19] Y. Yao, J. Duan, K. Xu, Y. Cai, Z. Sun, and Y. Zhang, "A survey on large language model (LLM) security and privacy: The Good, The Bad, and The Ugly," *High-Confidence Computing*, vol. 4, no. 2, p. 100211, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S266729522400014X

[20] F. N. Motlagh, M. Hajizadeh, M. Majd, P. Najafi, F. Cheng, and C. Meinel, "Large language models in cybersecurity: State-of-the-art," *arXiv preprint arXiv:2402.00891*, 2024.

[21] J. Rando, F. Perez-Cruz, and B. Hitaj, "PassGPT: password modeling and (guided) generation with large language models," in *European Symposium on Research in Computer Security*. Springer, 2023, pp. 164–183.

[22] X. Su, X. Zhu, Y. Li, Y. Li, C. Chen, and P. Esteves-Verissimo, " PagPassGPT: Pattern Guided Password Guessing via Generative Pretrained Transformer ," in *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2024, pp. 429–442. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/DSN58291.2024.00049